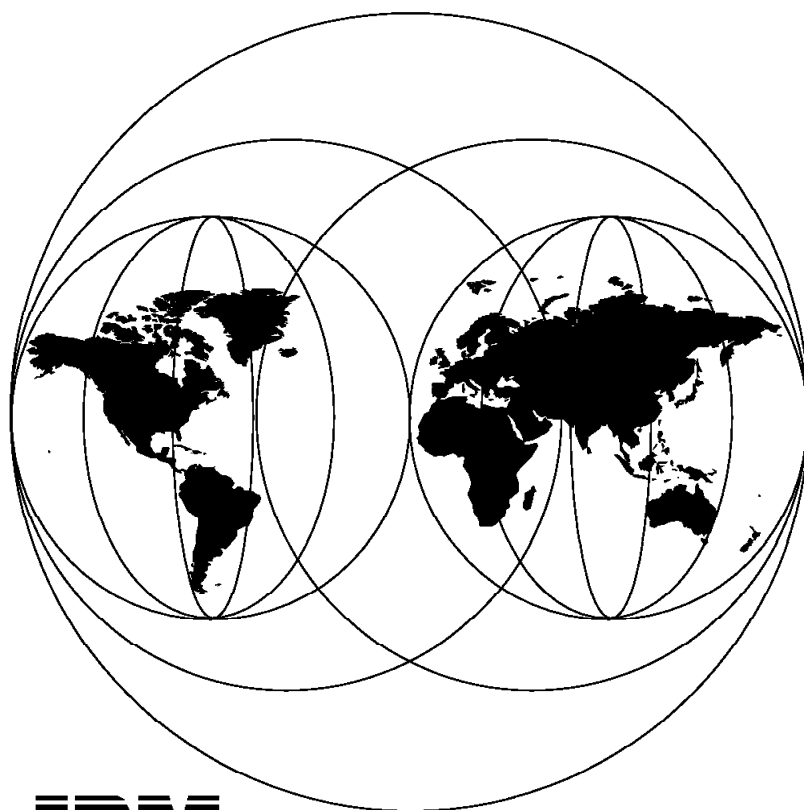


AS/400 Applications: A Fast and Easy Way to Install, Set Up and Work with VRPG and CODE/400 (ADTS CS)

January 1996



IBM

**International Technical Support Organization
Rochester Center**



International Technical Support Organization

SG24-4841-00

**AS/400 Applications: A Fast and Easy Way to Install,
Set Up and Work with VRPG and CODE/400 (ADTS CS)**

January 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, "Special Notices" on page 227.

First Edition (September 1996)

This edition applies to IBM VRPG Client Version 3, 5763-VR1, IBM Cooperative Development Environment for OS/400 and OS/2 Version 3, 5763-CD1 and IBM Application Development ToolSet Client Server for OS/400 Version 3 Release 1, 5763-CL1; Version 3 Release 6, 5716-CL1.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JLU Building 107-2
3605 Highway 52N
Rochester, Minnesota 55901-7829

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1996. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	xi
Tables	xiii
Preface	xv
Purpose of this Document	xv
Products	xv
How This Redbook Is Organized	xvi
Contents of the Diskette	xvi
The Team That Wrote This Redbook	xvii
Comments Welcome	xviii

Part 1. ADTS CS Installation and Run Time	1
Chapter 1. ADTS CS Installation	3
Communication Variables	3
Installation Scenarios	4
OS/400 Communications	4
OS/2 Communications Manager	6
PC Support/400 and Client Access/400 Installation	9
PC Support/400 and Client Access/400 for OS/2	9
Client Access/400 Optimized for OS/2	9
Client Access/400	10
Install ADTS CS	11
Chapter 2. ADTS CS Configuration	13
VRPG Client/2 Configuration	13
Define Server Information	13
Define Logon information for Servers	14
Define AS/400 Information	15
Dealing with Different Network IDs	16
When Things Go Wrong	17
CODE/400 Configuration	17
Chapter 3. ADTS CS Run Time	19
Override AS/400 Information	19
AS/400 Information Notebook Override	19
Edit the Remote Server Table	20
Override Logon Information	21
Chapter 4. Packaging and Installing	23
Packaging	23
Packaging Run-Time Code and Applications	23
Packaging a Component	25
Packaging a User-Defined Part	26
Installing Applications	26
Install the Run-Time Support on Client	26
Installing Packaged Application on Target Machine	28
Installing a Component Package	31
Installing a User-Defined Part	31

Chapter 5. Windows 3.1 Run Time	33
Before Installing Windows 3.1 Run Time	33
Windows 3.1 Run-Time Installation	33
General Considerations for Windows 3.1	34
Application Icons	34
Project File Names	34
Improving Performance	34
Windows 3.1 for WorkGroups	34
Printing	34
Restrictions	34
Other Notes	35
Memory Considerations for Windows	35
Client Access for Windows Considerations	36
Code Page	37
Network Options	37
PTF SF28451	37
Startup	37
QUSER Profile	37
Using Windows 3.11 for WorkGroups	37
Configuring DB2 for Windows Access	37
IPF_PATH Conflict	37
DB2/2 FixPak	37
Connect Reset	38
 Chapter 6. Win-OS2 Run Time	39
General Notes	39
Warp for Windows: Win32S	39
Installing WinOS2 Run-Time Support	39
Win32S Run-Time Support: Post-Install Checks and Note	40
WinOS2 Run Time - Settings Notebook	41
 Chapter 7. Pre-Installation Checks	43
AS/400 Environment	43
OS/2 and Communications Layers	43
OS/2 2.1	44
OS/2 Warp	44
LAPS	44
Warp Connect: MPTS	44
Communications	44
Client Access	45
Co-Existing Products	46
Just Before You Start Installing	46
 Chapter 8. Installation Problems and Errors	47
If WorkFrame/2 Phase 2 Failed	47
If Communications Manager/2 Failed	47
System Errors	48
Error Logging	49
 Chapter 9. After You Install	51
PTFs	51
Clean Up Old Debug Profiles	52
 Chapter 10. Some Errors That May Be Encountered	53
Before You Start	53

On the Client Side	53
On the AS/400 System Side	55
Error Messages with Message IDs	56
Error Messages Without Identifiers	59
Other Error Situations	62
The AS/400 Service Knowledge Base	63
RPG Compile and Run-Time Errors	64
Table of Run-Time Error Codes	65
 Chapter 11. ADTS CS Notes and Tips	67
Keeping in Touch	67
How to Obtain Corrective Service Diskettes (CSDs)	68
OS/2 Fix Packs	68
CM/2 Fix Packs	68
DB2/2 Fix Packs	68
VRPG Client and License Manager	68
Passwords	68
Running Two Versions of Application	69
Bitmaps and Icons	69
Debugging Windows Applications	69
Using Disk Caching	69
Blocking and Performance	69
Setting the VRPG Audio Alarm Option	70
Finding the Local LU Details Using CM/2	70
Test Directories	71
VRPG: Using Local Files	71
SQL Bind Files for DB2 Databases	72
Back Up and Restore	73
Backup Notes	73
Restore Notes	74
WorkFrame	74
Projects and Composite Projects	74
Renaming Projects	74
Tools	75
Moving Project Files	75
How to Determine the Service Level of Components	75
Removing Win32S	76
Specifying the AS/400 USERID and Password	76
Communications Troubleshooting	76
Running a Communication Trace on the AS/400 System	77
Running a Communications Trace on OS/2	78
Communication Trace on Windows 3.1	78
CODE Tips	80
The CODESRV Command	80
The EVFCCMD Command	81
The CODEEVNT Command	81
The CODEBRWS Command	82
Other Neat Things in CODE	82
Using ADM	82
Functions Supported	83
Error Feedback for CODE/400 with ADM Builds	83
DBCS Considerations	83
DBCS Code Page Support	84
 Chapter 12. GUI Designer Notes	85

Project Operations	85
Project Rename	85
Project New, Project Save	85
Entry Field Parts Settings Notebook Behavior	85
Specifying Range of Values for Entry Field	85
Changing Entry Field or Static Text Part to Numeric	85
AS/400 Information Notebook	86
Changing Action Links	86
Message Box After Restore	86
Message Description Length	86
Technical Descriptions	86
Windows Run Time: General Differences	87
Number of Parts	87
No Second-Level Help	87
No Drag-and-Drop	87
Win-OS2 Run Time: General Differences	87
Font Sizes	87
Events	87
Close	87
Destroy	88
Resize	88
Terminate on Close	88
Attributes	88
Color Attributes in Windows Run Time	88
Font Attributes in Windows Run Time	89
Height and Width in Windows Run Time	89
Mouse Focus in Windows Run Time	89
Multiplier Attribute	89
MultSelect and ExtSelect	89
Outline Font Attribute	89
SelectIDx and Groups	89
Display Size System Attributes	90
Parts	90
Audio	90
Check Box	91
Combination Box	91
Component Reference (CRF)	91
Container	91
DDE Client	92
Entry Field	92
Graphics	92
Graphic Push Button	93
Group Box	93
Image	93
List Box	93
Menu Item	93
Message Subfile	94
Multi-Line Entry	94
Push Button	94
Radio Button	95
Slider	95
Spin Button	95
Static Text	96
Subfile	96
Timer	96

Window (and Window with Canvas)	96
Event Attributes	97
VRPG Language and Compiler	98
Printer Files	98
Program Termination	98
Passing Parameters	98
HIVAL and *LOVAL	99
LPEX Editor	99
Program Specifications	99
Subroutines	99
Full View	100
Editing Action Subroutines	100
Writing Online Help	100

Part 2. Lab Exercises 101

Chapter 13. VRPG Client/2 Labs	103
Prerequisites	103
Overall Lab Configuration	103
Overview Exercise Description	104
Lab 1: Create a Simple VRPG Client/2 Application	104
Exploring the VRPG Client GUI Designer and Creating Action Subroutines	104
What You Should be Able to Do	104
Introduction	105
Starting the VRPG Client GUI Designer	105
Some LPEX Editor Features	120
Building the Application	126
Running the Application	127
Lab 2: Exploring VRPG Client GUI Designer and Accessing AS/400	
Databases	128
What You Should be Able to Do	128
Starting VRPG Client GUI Designer for Existing Project	128
Creating a User-Defined Part	133
Adding a New Window to Your Project	139
Adding More Logic to Your Project	156
Lab 3: Enhancing Your Application by adding Error Handling, Action Links, and Messages	162
What You Should Be Able to Do	162
Enhancing the Run-Time Behavior	163
Lab 4: Creating a component with Subfile and Application Help	172
What You Should Be Able to Do	172
Creating a Subfile	172
Lab 5: Using the Component Reference Part	179
What You Should be Able to Do	180
Creating the Hidden Field	180
Lab 6: Using Pop-up Menus and Notebooks	183
What You Should Be Able to Do	183
Creating a Pop-up Menu	183
Lab 7: Using the Container Part	186
What You Should Be Able to Do	187
Container Overview	187
Descriptions of Files Used in the Labs	191
DDS for Customer File	191
Customer File on AS/400 System	192

Contacts.TXT File	193
VRPG Source GUIDES2	194
VRPG Source for COMPLIST	198
VRPG Source for Container Example	201
README File	204
Setup	204
Installation and Setup	204
Contents of the Diskette	205
Help for AS/400 Upload Program	206
Source for AS/400 upload program	208
VRPG Client Lab Setup Instructions for the AS/400 System	213
Cleanup	213
CL Source for AS/400 Setup	214
CL Source for AS/400 Setup Command	215

Part 3. Information for Service Providers 217

Chapter 14. Service Offering	219
VRPG Client Installation and Configuration Service Offering	219
VRPG Client Education Service Offering	219
Inventory for Installation and Configuration	219
Company Description	220
AS/400 Systems Configuration (for Each System)	221
Network Configuration	221
Type of Service to Be Performed	221
System Software (for Each System)	221
IT Environment Overview Report	222
Initial Proposal	222
Sizing of Installation and Configuration Offering	222
Other Costs	222
Contracts	222
Contract Content	223
Statement of Work	223

Chapter 15. Post Services	225
Further Education or Support	225
Additional Service Offerings Out of Post Relationship	226

Appendix A. Special Notices	227
--	------------

Appendix B. Related Publications	229
International Technical Support Organization Publications	229
Online books supplied with ADTS CS	229
ADTS CS	229
WorkFrame	229
CODE/400	229
LPEX Editor	229
VRPG	230
IPF (Information Presentation Facility)	230
Other Useful Publications	230
ADTS CS	230
AS/400 System	230
OS/2 Warp	230
CM/2: Softcopy Provided	230

Communications Server	231
Personal Communications	231
Client Access/400	231
PC Support/400	232
ADM	232
Internet Resources	232
VRPG Home Page	232
IBM Public Information and Services Main Menu	232
Product Descriptions	232
Education	233
Service	233
General	233
How to Get ITSO Redbooks	235
How IBM Employees Can Get ITSO Redbooks	235
How Customers Can Get ITSO Redbooks	236
IBM Redbook Order Form	237
Glossary	239
Index	249

Figures

1.	Display Network Attributes	5
2.	Display Line Description	5
3.	OS/2 Communications Manager Open Configuration	6
4.	OS/2 Communications Manager Confirmation Windows	7
5.	OS/2 Communications Manager Configuration Definition - AS/400 System	7
6.	OS/2 Communications Manager 5250 Emulation Through Token-ring	8
7.	PC Support/400 Installation (PC to AS/400 Connection)	9
8.	Client Access/400 Optimized for OS/2 AS/400 Connection	10
9.	CA/400 Optimized for OS/2 Communications Setup - Local Node	10
10.	Enroll Client Access/400 Users	11
11.	Define Server Information	13
12.	Define Logon Information for Servers	15
13.	AS/400 Information Add Server Alias Name	16
14.	STRCODE Command	17
15.	Define AS/400 Information at Run Time	20
16.	AS/400 Information Notebook Override	20
17.	User Logon Information Override	21
18.	VRPG Client Status Indicator	105
19.	VRPG Client GUI Designer	106
20.	Displaying Menu Help	107
21.	Changing the Window Part Name and Title	109
22.	Icon View of Parts	111
23.	Selecting Parts	112
24.	Alignment Tools	113
25.	Changing Push Button Label	114
26.	Completed Customer Inquiry Window	115
27.	Opening a Window in the Project View	117
28.	Selecting an Event	117
29.	LPEX Editor for Exit Push Button Action Subroutine	118
30.	Insert with Prompt	118
31.	Code to Terminate the Program	119
32.	Code to Terminate the Program	120
33.	Positioning the Cursor to Column 7	122
34.	Only Comment Lines are Shown	123
35.	Row and Column Information	123
36.	Creating a Syntax Error	124
37.	Closing the Design Window	126
38.	Build Options Dialog	127
39.	Tree View of the Project	130
40.	Customizing the Tool Bar	130
41.	Moving a Tool Bar Push Button	131
42.	Customizing the Parts Palette	132
43.	Setting Image Part File Name	133
44.	Sizing the Image Part	134
45.	Change Fonts Setting	135
46.	Before aligning the new parts	136
47.	Adding a User-Defined Part to the Parts Catalog	137
48.	Create User-Defined Part Window	138
49.	Parts Palette with a User-Defined Part	138
50.	Window Settings Notebook	140
51.	Re-Sizing Static Text Part	141

52.	Design Window with Logo and Field Headings	142
53.	Specifying a Server Name	143
54.	Server List	143
55.	Logon Window	144
56.	Define Reference Fields - Field List	145
57.	Dragging a Reference Field	146
58.	Reference Field Page	146
59.	Window with Entry Fields	147
60.	Spacing Static Text Parts	149
61.	Alignment	150
62.	Left Align Entry Fields	151
63.	After Alignment	152
64.	Sizing CUSTNA Entry Field	153
65.	Re-Size CONTAC and CADDR	154
66.	Re-Size Window	155
67.	Final Window Design	156
68.	Using READ Operation Code	157
69.	Using the GETATR Operation Code	158
70.	Using the GETATR Operation Code	159
71.	AS/400 Information	160
72.	Displaying the Project Files	161
73.	Icon View of the Project	161
74.	Error Handling Code	163
75.	Code to Position Cursor	164
76.	Creating an Action Link	165
77.	Creating a Message in VRPG Client	166
78.	Adding the DSPLY Operation Code	167
79.	Logic to Change the Background Color	167
80.	Defining a Message Box	168
81.	Displaying the Message Box	168
82.	Using Message Substitution	169
83.	Defining Message Text in the Program	169
84.	Navigate with Action Link Window	170
85.	Debug Source View	171
86.	Specifying a Server Name	174
87.	Completed Window with Subfile	175
88.	Code to Fill the Subfile from the Database	175
89.	"F" Specification for CUSTOMER File	176
90.	Logic to START the Subfile Component	178
91.	Displaying the Pop-up Menu	184
92.	Defining the Local File	188
93.	Defining the Record Types	189
94.	Defining Definition Specifications	189
95.	Logic to Add Records to the Container	190
96.	DDS for Customer File	192
97.	DDS for Logical File CUSTOMLF	192
98.	DDS for Logical File CUSTOML3	192
99.	Data in CUSTOML3 File	193
100.	Data in CONTACT.TXT File	193
101.	Customer Inquiry Program	194
102.	Customer List Program	198
103.	Container Program	201
104.	Upload AS/400 Files Program	208
105.	CL Setup Source	214
106.	CL Setup Command Source	216

Tables

1. Communication Variables	4
--------------------------------------	---

Preface

Installation and configuration of VRPG Client with an AS/400 can cause long, frustrating hours of confusion for some users. Not because VRPG Client is hard to install, but because you need to coordinate the settings between all the products that you plan to install. You may need Communications Manager help, Client Access help, and VRPG Client help, and so on, to get the job done. Determining where to look, what you need to know, and what to do with the information once you have found it can prove difficult. This publication seeks to help the user not only in fast-tracking the install and setup phases, but also through hands-on exercises shortening the VRPG learning phase.

Since the install process is similar for CODE/400, this book serves a dual function in that it provides the extra information to quickly get CODE/400 operational.

Purpose of this Document

This document contains:

1. Information to assist in installation and configuration of VRPG Client and CODE/400.

This document covers the most common scenarios that you encounter.

2. Lab exercises to use as a tutorial or as structured lab exercises.

This document can be used alone or by a service provider to assemble a service offering.

Products

All the samples and lab exercises are based on OS/400 Version 3 Release 6 for PowerPC AS/400 series and OS/400 Version 3 Release 1 for non-PowerPC AS/400 series. The products used in this document are:

1. IBM VRPG Client Version 3, 5763-VR1 (V3R1M1)
2. IBM Cooperative Development Environment for OS/400 and OS/2 Version 3, 5763-CD1 (V3R1M1)
3. IBM Application Development ToolSet Client Server for OS/400 Version 3 Release 1, 5763-CL1; Version 3 Release 6, 5716-CL1.

The products listed in Step 1 and Step 2 are functionally the same as the products in Step 3, the only difference being in the directory structure. This difference is imposed by the way the install and PTF process work.

The information in this manual is designed to complement the install instructions and information on prerequisite products found in *Installing Application Development ToolSet Client Server for OS/400 Version 3*, SC09-2188-00.

How This Redbook Is Organized

This redbook contains 255 pages. It is organized as follows:

- Part 1, "ADTS CS Installation and Run Time"

This part contains information about installing and configuring ADTS/CS on the AS/400 system. It also has some diagnostic and error recovery information. The following chapters are included in this part:

- Chapter 1, "ADTS CS Installation"
- Chapter 2, "ADTS CS Configuration"
- Chapter 3, "ADTS CS Run Time"
- Chapter 4, "Packaging and Installing"
- Chapter 5, "Windows 3.1 Run Time"
- Chapter 6, "Win-OS2 Run Time"
- Chapter 7, "Pre-Installation Checks"
- Chapter 8, "Installation Problems and Errors"
- Chapter 9, "After You Install"
- Chapter 10, "Some Errors That May Be Encountered"
- Chapter 11, "ADTS CS Notes and Tips"
- Chapter 12, "GUI Designer Notes"

- Part 2, "Lab Exercises"

Part two provides you with labs that you can use for teaching this material or as a self-study tutorial. The included diskette contains files used in the labs. It also contains freelance presentations that can be used alone or in conjunction with the labs.

- Part 3, "Information for Service Providers"

Part three describes what you as a service provider can do with the information provided in this book.

Contents of the Diskette

The diskette contains the following:

File 4841SET.EXE

Run **4841set.exe** to create these files in directory 4841SET:

- README.TXT file with instructions on how to use the files on the diskette.
- CRTPRJ.CMD to create VRPG Client projects from an OS/2 command line.

File UPSAVF.EXE

Run **UPSAVF.EXE x:\ /P** to create these files in directory UPSAVF. The UPSAVF VRPG application allows you to create the AS/400 environment for the hands on LABS in Part 2 of this Redbook:

- The UPSAVF VRPG project files
- PF Customer
- Source PF QDDSSRC - DDS for customer file (physical and logical)

File GUIDES2.EXE

Run **GUIDES.EXE x:\ /P** to create project files in directory GUIDES2. It contains the source and runtime solution for Lab 1 through Lab 6.

File COMPLIST.EXE

Run **COMPLIST.EXE x:\ /p** to create project files in directory GUIDES2. It contains additional source solutions for Lab 1 through Lab 6.

File CONTAINR.EXE

Run **CONTAINR.EXE x:\ /p** to create project files in directory CONTAINR. It contains the source and run-time solution for Lab 7.

File 4841FLW.EXE

Run **4841FLW.EXE x:\ /p** to create files for two Freelance presentations in directory 4841FLW. Use the SHOW command to view these presentations. The SHOW command is part of this package.

- **OVERV04:**

VRPG Client, an introduction to Visual Programming with VRPG Client.

A set of foils to use as an overview and intro to the labs.

- **4841:**

Overview of installation and configuration.

File LAB7.EXE

Run **LAB7.EXE x:\ /P** to create directory LAB7. It contains the local PC files and icons used in Lab 7.

- Contacts.txt
- Manager.ico
- Employee.ico

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Rochester Center.

Jan Arbouw is an AS/400 Systems Specialist at the AS/400 Marketing and Support Organization in the Netherlands. His areas of expertise include technical support on AS/400 Client/Server Development, especially VRPG Client/2 and CODE/400. He has 27 years of experience within IBM, including development of IBM Business applications and supporting IBM AS/400 customers.

John Congerton is Senior Technical Consultant with IBM Business Partner Synergy Software in Melbourne Australia. He has 22 years of experience in IBM Midrange, including 9 years as a commercial DP manager and 13 years in consulting. His areas of expertise include database design and application development for the AS/400 system and for Client/Server.

Susan M. Gantner is a technical Advocate for AS/400 Application Development products and is located in the IBM Lab in Toronto. Susan specializes in application development and database on the AS/400 system and is a regular speaker at COMMON and Technical Conferences for AS/400 customers around the world. Susan has spent 21 years in the field of application development. Prior to joining IBM, she developed applications for corporations in Atlanta, GA., working with a variety of hardware and software platforms. In 1985, she joined IBM as a Systems Engineering Specialist for the System/38. She spent 5 years working in Rochester, Minnesota, supporting customers and IBMers around the world with database and programming challenges.

Cheryl Pflughoeft is an advisory programmer on the Object Oriented Team in AS/400 Partners in Development (PID). Her current job is to assist business partners with their migration to the Integrated Language Environment (ILE) and object-oriented technologies.

Prior to this, she was the coordinator for the PID early involvement programs for ILE C, ILE RPG, and ILE COBOL.

Cheryl started with IBM Rochester programming in-house manufacturing and warehouse applications on the S/3. She was part of the team that converted existing RPG applications to the S/38. Cheryl then moved on to the development team for Query/36 and Query/400 products, later becoming team leader for the Query/400 product.

Claus Weiss is a technical planner for Visual Tools in the IBM Toronto Lab. Currently he is working with the VisualAge for RPG development group on future releases of this product. Claus is an Industrial Engineer and also has a degree in Informatik from the University in Hamburg Germany. He joined IBM Germany as an SE for Midrange systems 18 years ago. He then moved to Toronto and joined the IBM Laboratory as a developer for the S/36 to S/38 migration aid. Claus then worked in the Design Control Group for the initial release of AS/400 AD tools. He also was involved with several AD tools as a planner in the AS/400 AD area.

Thanks to the following people for their invaluable contributions to this project:

Neil Willis
International Technical Support Organization, Rochester Center

Larry Schweyer
IBM Canada, Toronto Laboratory

Jon Paris
Technical Advocate, AS/400 Application Development Products, Toronto Lab

George M. Yeung
IBM Canada, Toronto Laboratory

Kim Doyle
IBM Canada, Toronto Laboratory

Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

Your comments are important to us!

Part 1. ADTS CS Installation and Run Time

This first part of this section guides you through setting up the communications environment on your workstation and the AS/400 system. It also provides tips on configuring ADTS/CS.

The ADTS CS run time is discussed as well as the packaging and installation of a VRPG Client application on OS/2 and Windows 3.1 platforms.

The remaining chapters in this part contain descriptions of error scenarios that have been encountered during installing and using ADTS CS. Recovery from these errors is also discussed.

A collection of notes and tips for ADTS CS from various sources concludes this part of the redbook.

Chapter 1. ADTS CS Installation

The installation of Application Development ToolSet Client/Server (ADTS CS) is not a difficult task to perform. You simply enter the install command and answer a few prompts. The complexity is in the combination of products involved in setting up a complete development environment for ADTS CS. All of the products involving OS/400, OS/2, Client Access/400, CODE/400, and VRPG Client/2 must be linked together before all development functions work well. After the installation of all products, some important configuration has to be done for the VRPG client side.

The installation of the products is explained in their respective installation documentation. If you follow these instructions, the installation should be easy. However, during the installation, you have to enter some variables. Because of the difference in terminology between the products, it can be confusing as to what to enter at certain prompts. Entering the wrong names for these variables can result in an ADTS CS development environment that partially works or does not work at all and it can be a time consuming job to find the error. This chapter explains what to enter during the installation steps. It is not exactly the same in all circumstances because there are many exceptions and different LAN environments.

For more information about what to do in unexpected situations, see Chapter 8, "Installation Problems and Errors" on page 47.

The physical installation of the environment consists of a number of steps:

- Install ADTS CS on the AS/400 system or on a LAN server. This step is not required if the ADTS CS client code installation is directly from a CD ROM.
- Install OS/2 on the workstation that is used for the ADTS CS development.
- Install the Communications Manager software and set up the configuration for the communication to the AS/400 host system.
- Install Client Access/400 client code on the workstation and configure the connection to the AS/400 system.
- Install ADTS CS features CODE/400 and VRPG on the workstation.
- Start CODE/400 and define the communication settings.
- Start VRPG and define the required communication settings.

Communication Variables

During the installation of OS/2 Communications Manager, Client Access/400, and VRPG Client/2, you have to enter a number of variables. It is not always clear what to enter for the variable names. The wrong names entered can lead to unexpected results.

The following table can be used as a reference for names that are used and which names must be entered on other displays. It shows the names used in the examples. You can fill in the names you are using. The character in the first column refers to the character used in the images of the displays.

<i>Table 1. Communication Variables</i>			
	Variables Description	Example Names	Your Names
A	AS/400 Local network ID	CAIBMOML	
B	AS/400 Local control point name AS/400 system (for connection)	TORAS016	
C	AS/400 Local adapter address LAN destination address	400011528646	
D	OS/2 Configuration name	AS400	
E	Local node name PC location name	OMXNZP	
F	Mode name	QPCSUPP	
G	PC location name (alias) Local node alias name	5250LU	
H	System to connect to (alias) Partner LU alias Server name Remote location	5250PLU	
I	User profile	VRPGDEV1	
J	Server Alias	AS400DEV	

The defaults are used for all other variables in the installation and configuration examples.

Installation Scenarios

The environment for the first scenario is:

- Token-ring local area network
- OS/2 Warp connect
- OS/2 Communications Manager
- Client Access for OS/2 (updated from PC Support/400)

The environment for the second scenario is:

- Token-ring local area network
- OS/2 Warp connect
- Client Access/400 Optimized for OS/2
- Communication Managers from Client Access/400

OS/400 Communications

You need some information from the AS/400 system when configuring communications on your workstation. For retrieving the OS/400 communication information, use two commands.

Use the DSPNETA command on the AS/400 system to find the names for:

- Local Network ID
- Local control point name

Figure 1 on page 5 shows the display that gives this information.

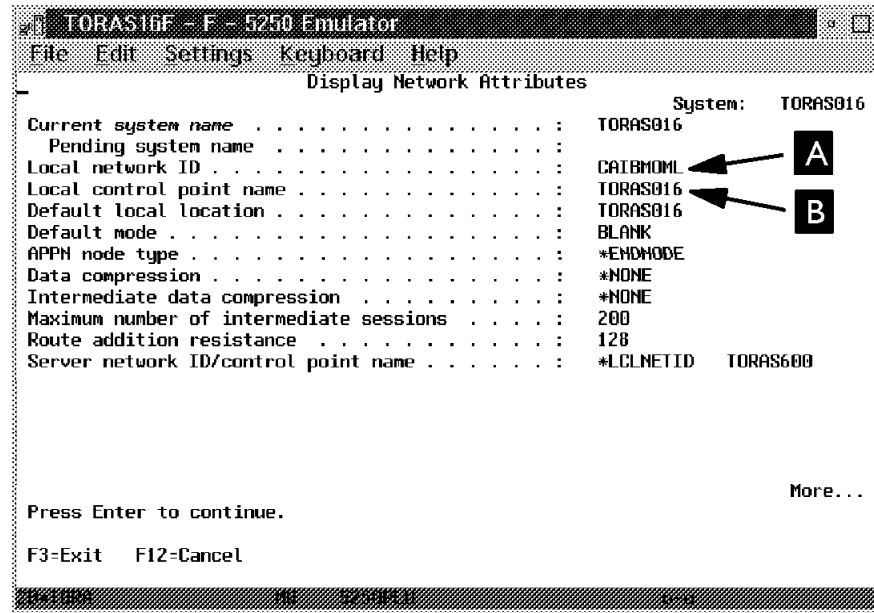


Figure 1. Display Network Attributes

Use the WRKLIND command to work with all the AS/400 line descriptions. In this example, the line description for the token-ring local area network (*TRLAN) was selected. Display the selected line description to show you the local adapter address of the LAN-card. Figure 2 shows the display for the DSPLIND command for the line description.

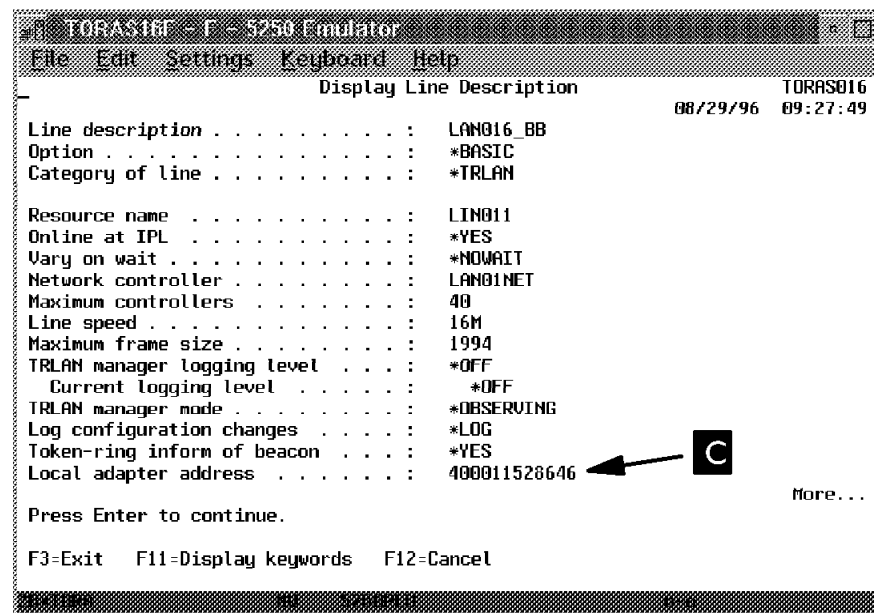


Figure 2. Display Line Description

OS/2 Communications Manager

After installing OS/2 and Communications Manager/2, you have to create a configuration file for the connection to the AS/400 host systems. Start Communications Manager Setup and enter a name for the configuration (in the example, this is AS400). Instead of Communications Manager/2, you can also use Client Access/400 Communications Manager. Figure 3 shows the Open configuration window.

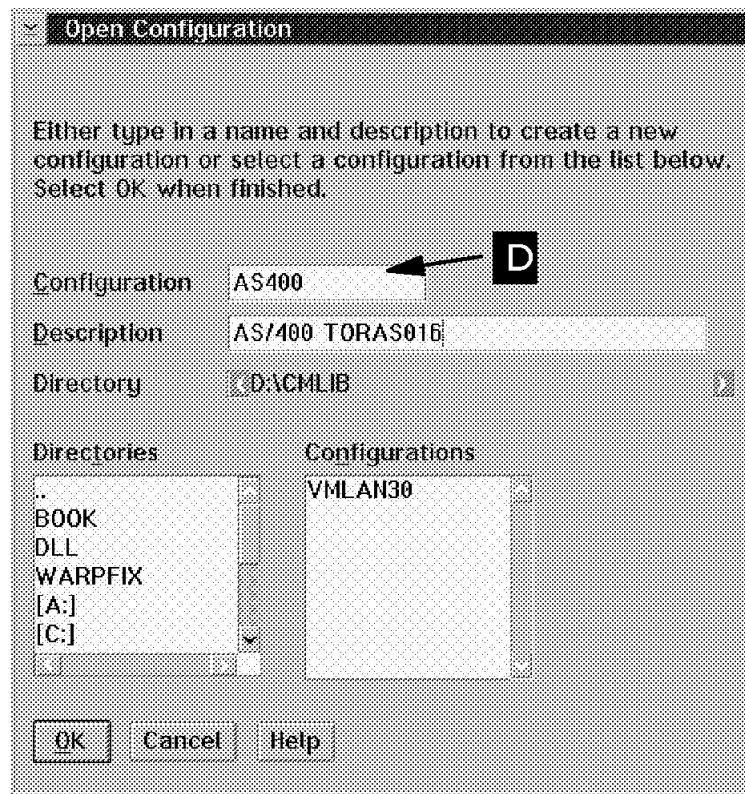


Figure 3. OS/2 Communications Manager Open Configuration

After pressing the OK pushbutton, confirm with Yes on the windows "create" and "used for this workstation" (see Figure 4 on page 7).



Figure 4. OS/2 Communications Manager Confirmation Windows

The first window for the new configuration asks for the type of connection. In the example, 5250 emulation through token-ring is chosen. On the Communications Manager Configuration Definitions - AS/400 window, select the type of connection you have and press the configure button (see Figure 5).

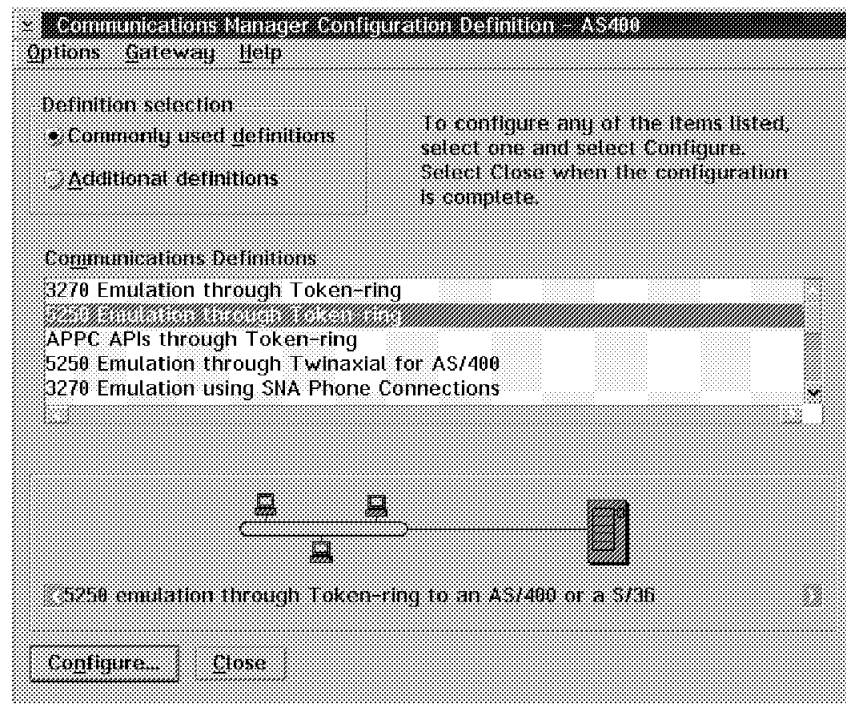


Figure 5. OS/2 Communications Manager Configuration Definition - AS/400 System

The most important window for the communications setup is the next window. Figure 6 on page 8 shows the 5250 emulation configuration window where you enter the variables that are required for setting up the communication.

- The network ID should be the same name as the local network ID on the AS/400 host.

Note: If your workstation is already configured with a different network ID and your network topology does not allow this ID to be changed to the name used on the AS/400 system, you have to change your ICF file. See “Dealing with Different Network IDs” on page 16 for tips to do this.

- The local node name is the name for this workstation as it is known by the network. Give each workstation a unique name and follow the standards used by the network.
- The local node ID is not important for the VRPG configuration and you can use the default unless your network configuration requires a certain value.
- The partner LU name is a combination of the network ID and the control point name from the AS/400 system. These names are connected by a period. The Communications Manager creates a combined name for this PLU alias name. The default alias name is 5250PLU and is used in the example.
- For mode name, our recommendation is to use QPCSUPP, but any other mode can be used as well as long as it exists on the AS/400 system.
- Enter the LAN destination address that is found using the DSPLIND command for the line.
- Enter the number of terminal and printer sessions you want to have. For the terminal session, specify at least one; two sessions are recommended.

5250 Emulation through Token-ring

Network ID: CAIBMOML **A**

Local node name: OMXNZP **E**

Local node ID (hex): 05D 00000

5250 information

Partner LU name: CAIBMOML.TORAS016 **A B**

Mode name: QPCSUPP **F**

Host type information: ☒ AS/400 ☐ S/36

☐ Connection is to a network node server

Connection information

LAN destination address (hex): 400011528646 **C**

Number of terminal sessions: 2

Number of printer sessions: 0

OK Advanced... Cancel Help

Figure 6. OS/2 Communications Manager 5250 Emulation Through Token-ring

After entering the desired information, press the OK pushbutton and close the configuration setup. The validation should not give any errors. The

configuration files are created. Use this configuration to start the communication and emulation sessions.

PC Support/400 and Client Access/400 Installation

Client Access/400 (or PC Support/400) is required for downloading the ADTS CS code from the AS/400 system as well as for PTF support. Install the product using the installation instructions.

PC Support/400 and Client Access/400 for OS/2

PC Support asks for the alias names of the local system and the remote system. Enter the correct names at the PC to AS/400 Connection prompt. When the defaults values are used, enter 5250LU for the PC location name **G** and enter 5250PLU for the name of the system to connect to **H**. Otherwise, enter the alias names you specified in OS/2 Communications Manager (see Figure 7). Alias names are case sensitive; therefore, we recommend that you always use uppercase characters for alias names.

Note: A word of advice; from our experience, ALIAS names are causing the most problems in the communications setup for VRPG. Be careful when you use ALIAS names. Make sure they are used exactly the same with letters having the same CASE in all of the different setup windows.

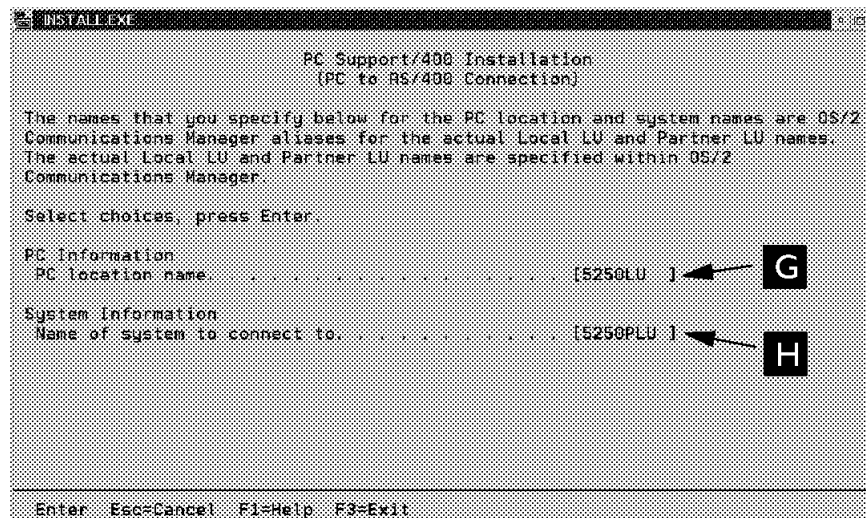


Figure 7. PC Support/400 Installation (PC to AS/400 Connection)

Client Access/400 Optimized for OS/2

Client Access/400 Optimized for OS/2 installation has a few prompts for the names of the network and the systems to connect. At the AS/400 Connection prompt (see Figure 8 on page 10), enter the name of the network and the name of the system. This information is found on the AS/400 system using the DSPNETA command.

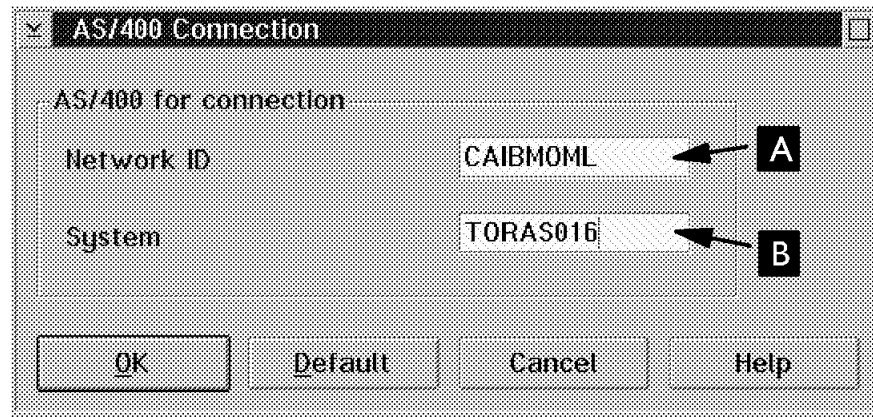


Figure 8. Client Access/400 Optimized for OS/2 AS/400 Connection

When asked for the communication setup, enter the names for:

- AS/400 Network ID
- Local node name of PC

See Figure 9.

Note: If your workstation is already configured with a different Network ID and your Network topology does not allow you to change this ID to the name used on the AS/400 system, you have to change your ICF file. See “Dealing with Different Network IDs” on page 16 for tips to do this.

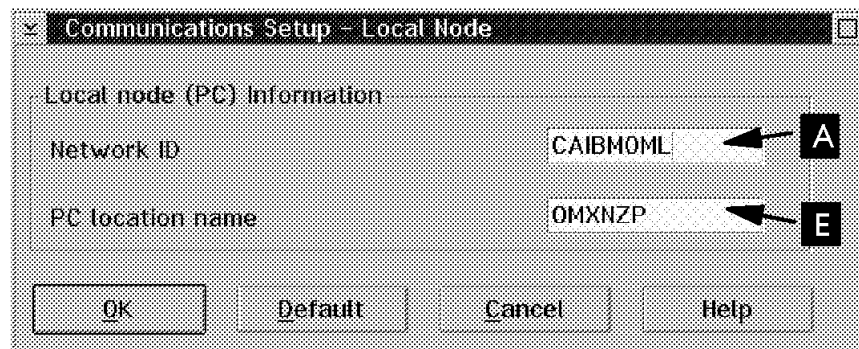


Figure 9. CA/400 Optimized for OS/2 Communications Setup - Local Node

Client Access/400

Enroll the VRPG development user as a Client Access/400 user. We recommend that you create a new user profile on the AS/400 system. This user profile is used for CA/400 authorization and also within ADTS CS for other functions (see “VRPG Client/2 Configuration” on page 13). For the examples, the user profile VRPGDEV1 is used.

- Create a user profile (if necessary) on the AS/400 system with CRTUSRPRF.
- On the AS/400 system, enter the GO PCSTSK command.
- Select option 21.
- Enter the user profile name and the USERID/address information (see Figure 10 on page 11).

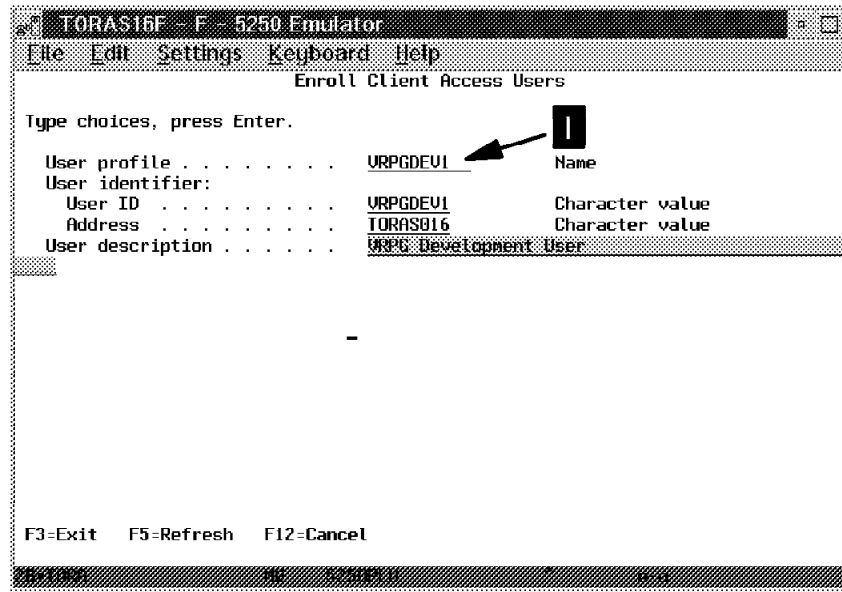


Figure 10. Enroll Client Access/400 Users

Install ADTS CS

After completing these steps, ADTS CS can be installed on the workstation. The installation is described in *Installing Application Development ToolSet Client/Server for OS/400 Version 3*, SC09-2188. Before you install VRPG or CODE/400, read Chapter 7, "Pre-Installation Checks" on page 43 to check if OS/400 and OS/2 environments are at the required level. If unexpected errors occur during installation, read Chapter 8, "Installation Problems and Errors" on page 47 for possible causes and how to correct the error. See also Chapter 9, "After You Install" on page 51 for more information about the installation of WIN32S support.

Chapter 2. ADTS CS Configuration

Both VRPG Client and CODE/400 configurations are discussed in this chapter.

VRPG Client/2 Configuration

When you need a server (AS/400 host system) while you are developing your application or running your application, you must ensure that VRPG has enough information to start the communication jobs and access the AS/400 data. Before accessing the host, you must define at least three types of information to VRPG.

1. Define the server using the Define Server window.
2. Define the name of the user profile using the Define Logon Information for Servers window. If you do not provide this information here, you are prompted for a USERID and password every time a connection to the AS/400 system is established.
3. Define additional AS/400 information using the Define AS/400 Information window.

Define Server Information

Start the VRPG development environment. From the project pull-down menu in the VRPG project window, select the "Define Server" option. Select "Add server information" and enter the name of the server to connect to. Check and select AUTOSTART for the server. If you do not select autostart, you always have to start the server using the STRCODE command on the AS/400 system. Figure 11 shows the window for entering the server name. After entering server name and selecting autostart, press the OK pushbutton.

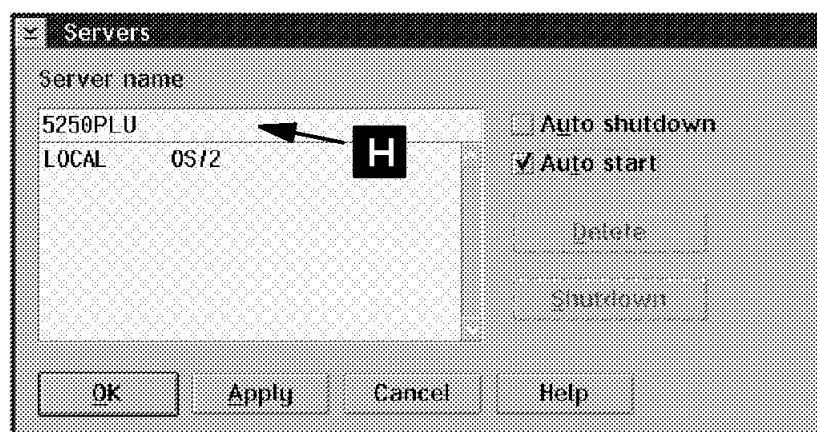


Figure 11. Define Server Information

Define Logon information for Servers

VRPG must have the name of a user profile to work with. Define the name of a user profile at the pull-down menu option "Define Logon information for Servers". If it is not defined here, VRPG communication jobs ask for it when needed. The information from the user profile is used at the startup of a server job. When you change anything regarding the user profile, you must first shut down the server before the new information can be used. This user profile is important for the following three reasons:

1. With VRPG Client/2, it is possible to access objects (files, programs, data areas) on the AS/400 system. You can only access objects on the AS/400 system with the right object authority. The user profile defined in the logon information for servers is used to access the objects on the AS/400 system.
2. One of the installation steps of ADTS CS is the creation of a customized Intersystem Communication Function file (ICF device file). Each ADTS CS development user must have an ICF device file in their own library. The file must reside in a library that appears in the user's library list. The library list is defined in the job description of the AS/400 job.
3. VRPG must be able to find the objects (that is, data files) on the server. To accomplish this, you have different options.
 - Use a library list. You can associate a library list with a job description on the AS/400 system. This job description can then be associated with a user profile. When starting a server with this user profile, the AS/400 server job contains the correct library list.
 - Define all objects with qualified names within VRPG. This can be done through the project pull-down menu option "Define AS/400 information". Select "Add a definition" for each object giving it an alias name.
 - At run time, call the QCMDDDM program to change the library list of the DDM service job. This ensures that the library containing the AS/400 database files is present in the DDM job's library list.

It is recommended to create a separate library for each developer that is only used for ADTS CS information. Place the ICF device file for the SNA communication in this library. Create a job description in this library that contains all of the libraries that are accessed. Make the library with the ICF file the first library in the user part of the library list to prevent using another developer's ICF file.

Figure 12 on page 15 shows the window for entering the logon information. After entering the information, press the OK pushbutton.

Figure 12. Define Logon Information for Servers. Add an AS/400 User.

Define AS/400 Information

During the development of your application, you can use the AS/400 Information notebook to define the following AS/400 information:

- Server alias names
- File alias names
- Program alias names
- Data Area alias names
- Commitment control

The AS/400 information notebook has three functions:

1. Define AS/400 objects with their fully-qualified AS/400 names. When you do not use the users profile library list, you have to define here where the VRPG communication job can find the objects on the server.
2. Provide an alias name for an AS/400 object. Instead of using the object name as it is called on the AS/400 system, you can give it another name and use this name in your VRPG projects.
3. Prepare AS/400 override information that is used at run time by the Remote Server Table (.RST file). The information in the .RST file is used during run time and gives you the flexibility to perform overrides when the run-time workstation is connected to another AS/400 system. See Chapter 3, “ADTS CS Run Time” on page 19 for information on how to override server and objects at run time.

Only the server alias name entry is required in VRPG; the other entries are optional. Without this information, the VRPG compiler is not able to resolve external descriptions so make sure that this information is supplied in the Remote Server Table. It is not necessary to define all of the objects in the AS/400 information notebook; only define them when you do not want to make use of the user’s library list. Use the option “Define AS/400 information” from the project pull-down menu to add a server alias name for the remote AS/400 server.

- Enter a name for the server alias. You can choose any name you want. This name is used in the Remote Server Table to assist in overrides at run time.
- The Remote location name must be the server PLU alias as defined in the Communications Manager.

Figure 13 shows the window for adding the server alias. After entering server names and descriptions, press the OK pushbutton.

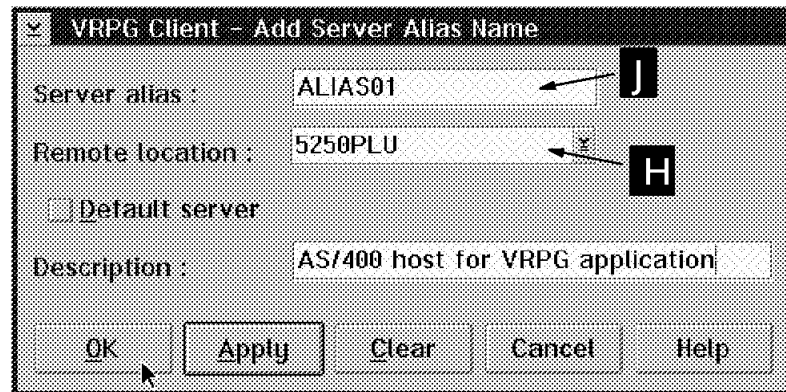


Figure 13. AS/400 Information Add Server Alias Name

Dealing with Different Network IDs

If you are dealing with different Network IDs for your AS/400 system and your workstation, this section has some useful information for you.

To work with the CODE server, you need an ICF file EVFCICFF that deals with ICF programs on your workstation. This ICF file has a default for the Remote Network ID of *NETATR, which means it tries to connect to a workstation in a network that has the same Network ID as your AS/400 system. This causes a problem because, in this case, your workstation is located in a different network. If you try to start the CODE server, it tries to connect to your workstation but fails because it is pointing to the wrong network.

You need to change the ICF files on the AS/400 system for all users that have a workstation in a different network than the network the AS/400 system is connected to.

Use the CHGICFDEVE command on your AS/400 system, specify the library your EVFCICFF ICF file is located in, specify the program device CODE, and change the RMTNETID parameter to the name of the network your workstation is located on. This change now points the ICF file to the correct network your workstation is located on.

Now the CODE server can be started and connects to your workstation.

Installing Application Development ToolSet Client/Server for OS/400 Version 3, SC09-2188 has additional information on this topic.

When Things Go Wrong

The installation and configuration for VRPG Client/2 is completed after you have defined server information, server logon information and the AS/400 Information notebook. You are now ready to create a VRPG program. Working in a client/server environment means you need to have a connection between the workstation and the host. For different reasons, this communication can be disconnected. The error message sometimes is a general message not telling you where to look. The error can be at the workstation side or at the host side. For more about communication errors and how to solve these, see Chapter 10, "Some Errors That May Be Encountered" on page 53.

CODE/400 Configuration

CODE/400 does not have a function to make the connection to the AS/400 host automatically. The connection to the AS/400 system must be initiated by the host system and not by the client. In a 5250 emulation session, enter the STRCODE command with the correct parameters. When the connection is made, you hear three beeps on the workstation and the window shows you the CODE/400 development window. This means that one 5250 session is locked during the time you are developing with CODE/400. The STRCODE can also be used for pre-start of a VRPG server session, but is not required. Figure 14 shows the window with the parameters to enter for the STRCODE command on the AS/400 system.

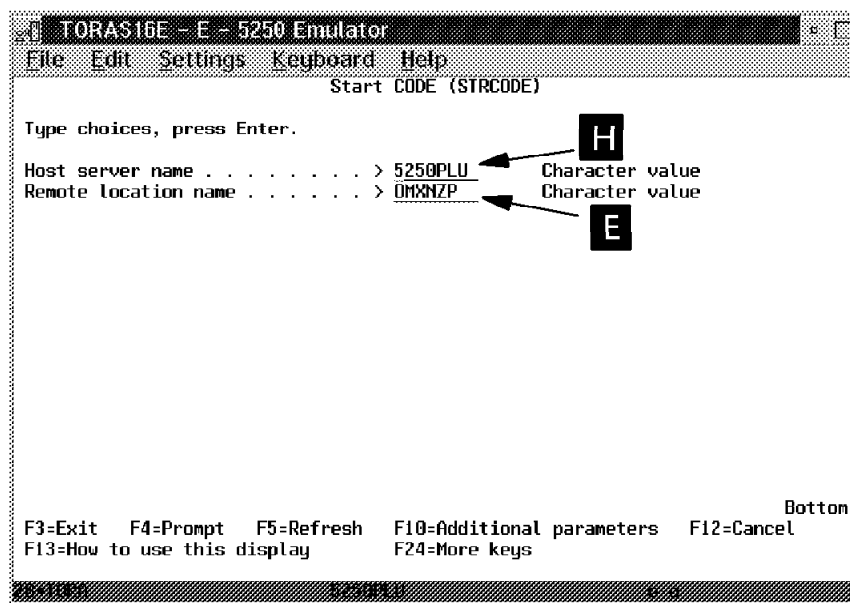


Figure 14. STRCODE Command

Chapter 3. ADTS CS Run Time

During the development phase, an application can be executed and tested within the VRPG development environment. When the application is ready to be distributed to the users, it can be packaged and installed on their workstations. Start the packaging function for a project in the GUI Designer, or start it directly from its icon in the VRPG Client/2 folder. The packaging utility collects all of the files necessary to run the application. For the run-time environment, you need the application code and run-time code. Select the packaging of the run time only the first time you package applications for a workstation because the run time is the same for all applications. Install the run time on the user's workstation using the VRRINST command. This installation must be completed with a re-boot before the VRPG application can be installed using the VRAINST command.

For more detailed information about packaging OS/2, WIN-OS2, and Windows 3.1 run times, see Chapter 4, "Packaging and Installing" on page 23.

Override AS/400 Information

When developing your application, use the AS/400 information notebook in the GUI Designer to define the information required to connect object alias names in your application and the server where they are located. When you fill the various pages on the AS/400 notebook, the GUI Designer updates the Remote Server Table (.RST file). In the run-time environment, you can override the .RST file information in order to connect object aliases to other systems and objects.

There are three ways to do overrides for the run time:

1. Before packaging the application, update the AS/400 information notebook in the GUI Designer. In this case, the packaging function makes a copy of the updated Remote Server Table and installs this in the application directory on the workstation.
2. On the run-time workstation, open the run-time folder (default is VRPGRT) and in this folder, open the BIN folder. Double-click on the icon for the AS/400 information notebook update program (FVDERST.EXE) to start the notebook update and define the overrides for the run time. This function is not available in VRPG release 1.
3. Update the Remote Server Table with an editor and change or add the DEFINE entries directly.

AS/400 Information Notebook Override

After starting the FVDERST program, you first see a window with a list of all the applications installed on the workstation. Select the application you want to change the overrides for and click on the OK push button. See Figure 15 on page 20.

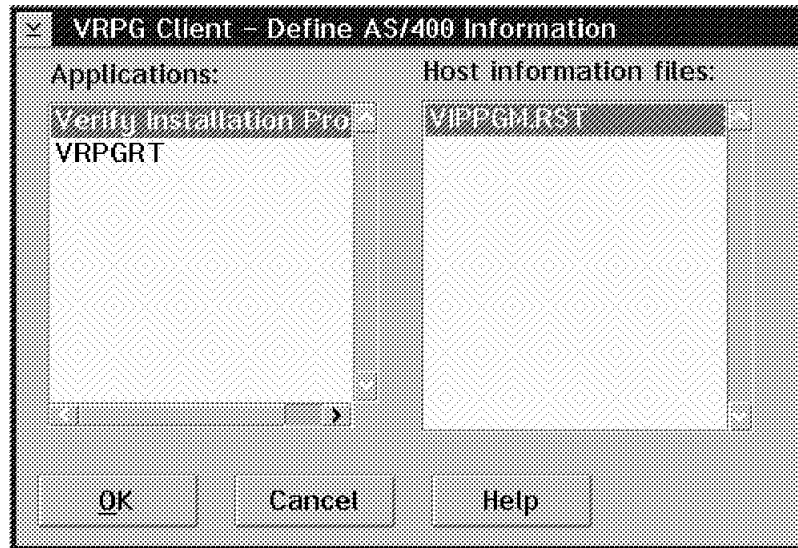


Figure 15. Define AS/400 Information at Run Time

The user's workstation can be connected to a different AS/400 host system (that is, the production system) than the development workstation and the Communications Manager definitions can be set up differently. Therefore, you must specify the correct AS/400 PLU alias name for this user in the field "Remote location". Do not change the content of the field "Server alias" even though the window title may appear to suggest that you do it. See Figure 16.

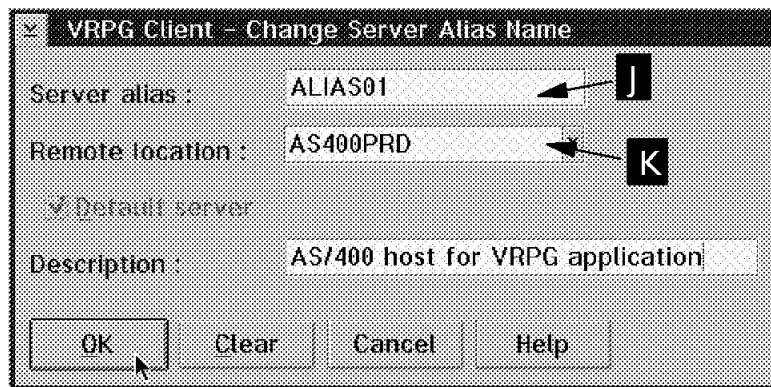


Figure 16. AS/400 Information Notebook Override

Edit the Remote Server Table

The Remote Server Table contains all of the alias information from the AS/400 Information notebook. With a PC editor, you can update the DEFINE entries. The following example shows entries in the RST file:

```
DEFINE_SERVER SERVER_ALIAS_NAME(ALIAS01) J
              REMOTE_LOCATION_NAME(5250PLU) H
              TEXT('AS/400 host for VRPG application')
```

If the user is connected to another AS/400 host, you can change the contents of the REMOTE_LOCATION_NAME parameter.

Another example of entries in the Remote Server Table is:


```

DEFINE_SERVER SERVER_ALIAS_NAME(ALIAS01) J
              REMOTE_LOCATION_NAME(5250PLU) H
              TEXT('AS/400 host for VRPG application')
DEFINE_FILE FILE_ALIAS_NAME(CUSTOMER)
              REMOTE_FILE_NAME(ORDLIBR/CUSTL1(MEMBER)
              SERVER_ALIAS_NAME(ALIAS01) J
              TEXT('Customer file')
DEFINE_FILE FILE_ALIAS_NAME(PRODUCT)
              REMOTE_FILE_NAME(ORDLIBR/PRODL1(MEMBER)
              SERVER_ALIAS_NAME(ALIAS01) J
              TEXT('Product file')

```

You can see the function of the SERVER_ALIAS_NAME. When you change the name in the REMOTE_LOCATION_NAME in the DEFINE_SERVER entry, you do not have to change all DEFINE_FILE entries. Because of the SERVER_ALIAS_NAME, all object aliases are now connected to the new server in the REMOTE_LOCATION_NAME.

Override Logon Information

When the VRPG run-time environment accesses an AS/400 host for data, you have to define a user profile that can be used by the VRPG communication jobs. The same rules apply here as in the development environment. The user profile must have the correct AS/400 authorizations and the library list must be set up correctly. When accessing data, VRPG first uses the Remote Server Table as a reference to find the objects. If not defined in the RST, VRPG uses the library list connected to the user profile. A recommendation is to create an individual profile for each user. On the run-time workstation, open the run-time folder (default is VRPGRT) and in this folder, open the BIN folder. Double-click on the icon for the AS/400 Logon Information update program (FVDEPW.EXE) to start the password override for the run time. This function is not available in VRPG release 1 and must be defined in the development environment before packaging the application. Figure 17 shows the window for the user profile override.

The screenshot shows a dialog box titled "VRPG Client - Add AS/400 User". It has the following fields and controls:

- User name :** A text box containing "USEROE".
- ✓ Password required**: A checked checkbox.
- Password :** A text box containing "*****".
- Password : (for verification)**: A text box containing "*****".
- Remote location :** A text box containing "AS400PRD". A black box with the letter "K" is overlaid on the right side of this field.
- Description :** A text box containing "Order Entry VRPG User".
- Buttons:** At the bottom, there are five buttons: "OK", "Apply", "Clear", "Cancel", and "Help".

Figure 17. User Logon Information Override

Chapter 4. Packaging and Installing

This chapter describes:

1. The packaging of VRPG run-time support, applications, components, and parts on a development machine.
2. The installation of all these packaged objects on a target machine ("Installing Applications" on page 26).

Packaging

Packaging has four meanings to VRPG:

1. Package the VRPG run-time code provided by IBM.
2. Application run-time packaging
3. Component packaging (toolkit function)
4. User-defined part packaging (toolkit function)

The corresponding installation processes for the target are:

1. VRPG run-time support install
2. Application run-time install
3. Component install (Toolkit)
4. User-defined part install (Toolkit)

Packaging Run-Time Code and Applications

If you have your own files (data files and any other files not created by VRPG) in the RT_OS2 directory, you must use the following rules for naming files. If you do not, you cannot package an application:

- a. The names for your files cannot be the same as the name of your VRPG Client/2 application.
 - b. The names of your files cannot have the same name or the same first two letters of the extension.
1. Clear the directory where you are going to put the package. The package function does not allow any files in the nominated target directory.
 2. Invoke the packaging utility using one of these methods:
 - Select the **Tools - Package** option from the GUI Designer.
 - Select the **Package** option from the project icon.
 - Select the Packaging Utility icon from the VRPG Client folder.

If the application is enabled for both OS/2 and Windows (or if you use the packaging utility), the Package window appears and you specify which OS you are creating a package for.

Either the Package OS/2 or Package Windows 3.1 window is shown.

3. Specify what you want to package:

- Application name:

This is the fully-qualified application project name (the name of its desktop representation). You can over-type it or use Find. When you specify an application project name, the title of the application is displayed for you.

- Package the application, or VRPG run-time support, or both.

By default, only the application is packaged, but you can send a VRPG run-time support package with it if you want.

4. Specify the application package information:

- **Target directory:** Where to put the package code. By default, A: is used.
 - If you package to a diskette, the target directory can only be the root directory of the diskette. There can be no subdirectory.
 - If you package on a LAN directory, that directory must not contain other files. If it does, you receive a message asking you to remove the data from the directory or specify another directory.
- **Title:** Your title for the application.
- **Number:** Your number for the application. (The number for IBM is 5763-CL1.)
- **Version:** Your version for the application. (The version might be 030101, which is edited to 03.01.01 in the Installation Utility displays. This number *controls* the behavior of the Installation Utility.
- **Feature:** Your feature for the application (optional).
- **Service level:** Your service level (optional).

Note: The title, number, and feature are mandatory. They are used to keep track of the version of the application. Enter this information carefully. If you package an update, make sure that you use the same title, number, and feature as before.

Select **Package** to begin. A progress indicator window is shown. Messages are displayed to tell you what labels to put on the various diskettes as you create them.

When packaging is completed, a completion message is displayed.

5. Specify the run-time package information:

If you specify that you want to package the run-time code, you must specify the target directory in the Package OS/2 run-time or Package Windows 3.1 run-time window. By default, A: is used. If you package to a diskette, the target directory can only be the root directory of the diskette. There can be no subdirectory. If you package to a LAN directory, that directory must not contain other files.

Select **Package** to begin the packaging. A progress indicator window is shown. When packaging is complete, a completion message is displayed. (You create the WIN32S1 and the WIN32S2 diskettes only if the target system is Windows.)

Package Contents

You find (at least) the following files in the packaging target directory, for an application named MYAPP:

- EVFDIDX.DE@ file is needed for PTF support.
- FVDIAPPI.DLL resource DLL handles messages.
- FVDIDELE.RSP response file says whether OVERWRITE FILES is allowed.
- MYAPP.DL@ is a compressed application component DLL.
- MYAPP.HL@ is a compressed application help (.HLP) file.
- MYAPP.ICF catalog file describes the name, number, and version of the application.
- MYAPP.OD@ compressed ODX defines the GUI parts.
- MYAPP.PKG packaging control file describes each object packaged.
- MYAPP.RS@ is the compressed version of the install response file.
- VRAINST.EXE is a copy of the common application install program.

You execute VRAINST.EXE in this directory to install the application. Alternatively, you can use the **Installation Utility** features.

Packaging a Component

This function is intended to assist developers, and is part of the IBM Toolkit in the VRPG Client folder. Its principal use is to copy or move VRPG project components to other drives or machines.

The source code, run-time directories, or both, can be packaged.

The directory into which the package code is delivered can be on diskettes, a local drive, a LAN drive, or an AS/400 shared folder.

As with the run-time packaging described previously, the *target* that you nominate for packaging the files into must be completely empty.

See “Installing a Component Package” on page 31 for the process to install the packaged component.

Two important notes to remember are:

1. This function is a bit serious about the word *component*. In VRPG (or at least Workframe) terminology, a component provides a .DLL, not an .EXE. If you use the function to package what *was* an .EXE project, you find a .DLL type when you install; that is, the first page of the project Settings notebook shows that the *target of project build* is yourname.DLL, where you originally had yourname.EXE.

You can change the deliverable back to an .EXE here if you go to the first page in the Settings notebook, change the extension of the build target from .DLL to .EXE, and re-build the project.

2. Unless you also select to package the project run-time code and install it also, you need to do a build to start using the installed code on another machine.

This is because a copy of the run-time components for a project has to be in the CODE\TEST directory (or WINTEST for a Windows build) of any development machine that tries to debug the project or run it in development mode.

This feature exists because of the fixed LIBPATH implementation in OS/2. The VRPG developers decided to create a run-time directory called TEST on the workstation when VRPG gets installed. This directory is also added to the LIBPATH and PATH statements in CONFIG.SYS during installation. This avoids the need to add a run-time directory to the LIBPATH every time a new VRPG project gets created. In a VRPG application build process, two copies of the run-time objects get created, one copy lands in the RT project directory, the other in the VRPG TEST directory. It is a good idea from time to time to clean up the TEST or WINTEST directories and delete old project files.

So, to duplicate a development project on another machine, you also have to duplicate the CODE\TEST and CODE\WINTEST entries for it. You can do this by rebuilding on the new machine, or by arranging your own copy of the project run-time components into the target CODE\TEST. We recommend that you rebuild, as this also tests that everything else required (.RST

entries, directories, LAN connections, folders, and local files) has been correctly established on the new development client.

If you decide to copy the files into the TEST directory instead of using a build, make sure that you unhide the files in the RT directory first.

- ATTRIB -h *.*

If you do not remove the hide attribute, some of the application files may be missing in the TEST directory and an attempt to run the application will fail.

Packaging a User-Defined Part

This function is intended to assist developers, and is part of the IBM Toolkit in the VRPG Client folder.

When user-defined parts are present in your parts palette or catalog, the toolkit "Package User-Defined Part" function is able to recognize them and package them for distribution to other development machines.

You can package to a diskette, local drive, LAN drive, AS/400 shared folder, or IFS directory. Just ensure that the diskette or directory is empty first, or the packaging is not allowed. You can package an unlimited number of parts in one run.

Consider creating a proper directory structure for user-defined parts as a centralized developer resource. Use a LAN directory, folder, or IFS directory.

When packaging is complete, the directory contains one .ODF (object definition) file for each part, plus a VRPGUPD.### file containing the details for install.

See "Installing a User-Defined Part" on page 31 for the process to install the packaged parts onto another ADTS CS workstation.

Installing Applications

You must:

1. Install the appropriate client run-time support (OS/2, Windows 3.11, or Win-OS/2).
2. Install the application package (component or part).

Install the Run-Time Support on Client

You *must* do this first before you deal with applications.

This only needs to be done once per client per release of the IBM run-time support.

You even have to do it for a development PC if you are going to install packaged applications on it.

A re-boot is required.

The procedure is:

1. Package the run time, or locate the existing package and transfer it to the client.

2. Change to the directory containing the packaged run-time support.
3. Execute the installation utility:
 - For OS/2, execute VRRINST.EXE from that directory.
 - For Windows 3.1, use File Manager to run SETUP from that directory.

Optionally, you can also specify these other parameters:

/A: Specifies the action. The default is /A:I to install the files. Specify /A:U to update the current version of the files. Specify /A:D to delete the files.

/L1 Specifies the fully-qualified file name of an error log.

The path typically points to a location on the LAN server so that an administrator has access to the logs in case of a failure. If the log already exists, the error message appends to it.

/L2 Specifies the fully-qualified file name of a history log.

The path typically points to a location on the LAN server so that an administrator has access to the logs in case of a failure. If the log already exists, the history information is appended to it.

/R Specifies the fully-qualified file name of a response file.

/X Specifies the action is run in unattended mode.

No progress indicators are shown. Messages are logged in the error log file and other messages in the history log file. If you use this parameter, you must create a response file before you run the unattended installation (see "Creating a Response File" on page 29).

Your CONFIG.SYS (for OS/2) or your AUTOEXEC.BAT (for Windows) is updated automatically.

You must enter or select the path for installing the run-time code. The default is c:\VRPGRT.

If you are installing the run-time code on the Windows system:

- Select whether or not you want to install the Define Server Logon and Define AS/400 Information icons; we suggest that you **do**.
- Install WIN32S if it is not already installed on the workstation.

4. When the installation is complete, re-boot the PC.

Install the Run-Time Support from LAN Server

If the run-time code or application is packaged on a diskette and you want to install from a LAN server, you must prepare the LAN server first. Basically, this involves copying the packaged information from the diskettes to a directory on the LAN server.

1. Create a directory for the run-time code on the LAN server.
2. Copy all run-time code files from diskettes VRPGRT1 through VRPGRTn to the run-time directory on the LAN server.
3. If you are installing on a Windows system:
 - a. Create two sub-directories in the run-time directory: DISK1 and DISK2.
 - b. Copy the contents of the WIN32S1 diskette to the DISK1 directory, and WIN32S2 diskette to the DISK2 directory.
4. Create a directory for the application on the LAN server.

5. Copy all application files from the diskettes to the application directory on the LAN server.

Note: You can use the following command to copy files:

```
COPY A:\* U:\path
```

where A: is the diskette drive letter or directory containing the files to be copied and U:\path is the directory where the files are to be copied to.

With the LAN server prepared, you are ready to install the run-time code and application from the LAN server.

Install the Run-Time Support Directly from Diskettes

1. Change to the directory containing the packaged application.
2. Execute VRAINST.EXE from that directory.
3. Select the drive and directory in which the application will install.
4. The application icon is placed on the desktop.
5. The Installation Utility may be used to manipulate the components from here on.

Installing Packaged Application on Target Machine

The run-time code must be already installed on a workstation before you can install an application.

Note: Do not install more than one application in the same directory.

1. For OS/2 local or LAN drive installations, obtain an OS/2 command window, set the current directory to the required VRPG package directory, and type VRAINST.
2. For Windows, WinOS2 local, or LAN drive installs, start the File Manager, select the VRPG package directory, and run SETUP.exe in it.
3. To install from diskettes, insert the first diskette into drive A:, and then:
 - For OS/2, enter A:\VRAINST.
 - For Windows, use the file manager to run A:\SETUP.

You can also specify other optional parameters (see "Install the Run-Time Support on Client" on page 26).

The CONFIG.SYS (for OS/2) or AUTOEXEC.BAT (for Windows) does not need to be updated.

4. Choose whether you want to overwrite existing files.
5. Enter or select the path for installing the application. The default is c:\packagename.
6. Select Install to begin the installation. The installation starts and the progress indicator is shown. If you are installing from diskettes, you are prompted to remove and insert additional diskettes as required (depending on the size of the application).
7. For Windows, select whether or not you want to install the Define AS/400 Information icon.

The Define AS/400 Information utility is installed with the application. Use this utility to maintain and update the names and location of AS/400 resources at run time.

Directory Contents

You find (at least) the following files in the installed target directory for an application named MYAPP:

- EVFDIDX.DES
- MYAPP.DLL (more DLLs if your application has multiple components).
- MYAPP.EXE
- MYAPP.HLP
- MYAPP.ODX
- MYAPP.PKG
- MYAPP.RST

In addition, the application icon is added to your desktop. It is set to execute your .EXE in the install target directory.

If this is all new to you, this is a good time to look at the contents of the .RST file, since it determines the run-time connections your program makes.

For instance, if you reference local files on the PC and you do not use the EXTFILE keyword in your F specs, they need to have file definition entries in the .RST file to tell VRPG where the files live. Otherwise, it creates them in the application run-time directory.

If the host your run time is going to talk to is different from the one your VRPG application was built on, you also have to change the .RST file. The Remote_Location_Name is the entry you have to change in the .RST file.

Creating a Response File

If you do not want to manually provide input during the installation process, you can prepare a response file to predefine the necessary information.

Two sample response files, VRPGOS2.RSP for OS/2 and VRPGWIN.RSP for Windows, are shipped with VRPG Client. You can edit the contents of each file to create a response file to suit your needs. Instructions are provided in the files.

The sample response files are in the /ETC subdirectory of the root directory where VRPG Client was installed. For example, if VRPG Client was installed in the CODE directory on your D drive, the files are in D:\CODE\ETC.

Maintaining Run Time and Application Code

From time to time, it is necessary to provide updated versions of the run-time code and applications. Your users can use the Installation utility to update or remove the run-time code and applications from their workstations.

To update the run-time code or VRPG applications using the Installation Utility, use the following steps:

1. Invoke the Installation Utility:
 - On Windows and WIN-OS2, start the Program Manager. Then select the Installation Utility icon from the Software Installer group window.
 - On OS/2, select the Installation Utility icon from the Software Installer folder on the desktop.

The Installation and Maintenance window is shown.

2. Select what you want to update, and select Update from the Action menu.

By default, a backup version is saved. You can choose not to have the backup version saved.

When you update the run-time code, your CONFIG.SYS (for OS/2) or your AUTOEXEC.BAT (for Windows) is automatically updated.

If you are updating the run time on Windows, you can choose to have the Define Server Logon and Define AS/400 Information icons installed. Similarly, if you are updating an application on Windows, you can choose to have the Define AS/400 Information icon installed. If you choose not to have the icons installed during the update, any icons that were installed with the previous version are deleted.

Note: You can update the run-time code or the application from the command line instead of the Installation Utility. The update command is identical to the installation command, except you must specify the update parameter, /A:U.

Removing Run Time and Applications

To remove the run-time code or VRPG applications using the Installation Utility, use the following steps:

1. Invoke the Installation Utility:
 - For Windows or Win-OS2, start the Program Manager. Select the Installation Utility icon from the Software Installer group window.
 - On OS/2, select the Installation Utility icon from the Software Installer folder on the desktop.

The Installation and Maintenance window is shown.

2. Select what you want to delete. Then select Delete from the Action menu.

Note: If you plan to delete both the run-time code and applications, you should delete the applications first because the Installation Utility is deleted along with the run-time code.

Note: You can delete the run-time code or the application using the command line instead of the Installation Utility. The delete command is identical to the installation command, except you must specify the delete action parameter, /A:D.

Embedded SQL

If your application has embedded SQL and is referencing a database to which your workstation does not have access, you have to re-bind your application to a database to which it does have access. To do this, you have to create a bind file through the compiler options when you create your application.

For example, assume you built your application and created your database package for database Types on workstation A. Also assume that your application does not have access to database Types on workstation A from workstation B. If you install your application on workstation B, and workstation B has database Typesx, you have to bind your application to that database using the bind file you created. In this example, you type the following command on an OS/2 command line:

```
sqlbind applic1.bnd typesx
```

where *sqlbind* is a DB2/2 command, *applic1.bnd* is the bind file created by the compiler when you built your application, and *typesx* is the database you want to access on workstation B.

Installing a Component Package

A VRPG component package created by the 'Component Packaging' toolkit utility is installed as a new project on another ADTS CS development machine by using the toolkit 'Component Install' function.

The decisions you need to make before running the install are:

1. Where is the project icon placed?
2. What source directory are the project sources files placed in?

If you packaged the run-time files as well as the source, they are placed in the RT_OS2 and RT_WIN subdirectories of the project directory you nominate.

As outlined in "Packaging a Component" on page 25, this utility currently delivers any VRPG project as a component that only contains a .DLL, even if the original was an application containing an .EXE. You have to alter the settings back to an .EXE suffix and rebuild on the new machine to get back from a component to an application.

Installing a User-Defined Part

User-defined parts intended to be added to the parts palette of an ADTS CS developer workstation must have been created in a dedicated directory or diskette by the toolkit User-Defined Part Packaging utility described in "Packaging a User-Defined Part" on page 26.

To install the parts, select the User-Define Parts Install icon in the VRPG Toolkit folder.

Give the directory name where the packaged part is found, and press the **Fill** push button to see the contents.

By default, all of the available parts are preselected for installation. You can install selectively by clicking on the list contents. Press OK when you are ready to install.

The selected parts are installed to the parts palette (and catalog) of the VRPG GUI Builder immediately.

Chapter 5. Windows 3.1 Run Time

The procedures are similar to those for the OS/2 environment, so the preceding chapter should be read and understood first, and can be consulted for the appearance and details of the dialogs.

Most of the detail here also applies to the Win-OS2 environment; however, you should also refer to the following chapter for specific comments on the Win-OS2 run-time environment.

For specific restrictions and variations related to parts, events, and attributes, refer to Chapter 12, "GUI Designer Notes" on page 85.

Before Installing Windows 3.1 Run Time

1. Ensure that TCP/IP for DOS is stopped.
2. Ensure that SHARE is running.

If either of these steps is omitted, the run-time installation may hang.

Windows 3.1 Run-Time Installation

Each workstation that is used to run installed VRPG applications under Windows requires this procedure to be performed *once*:

1. Open Windows Program Manager.
2. Run **Install.exe** from the directory into which you packaged the Windows run time.
3. Press **Continue** on the install prompt.
4. When prompted, type the directory into which the Window run-time support code should be installed.

This can be a LAN drive.
5. EPFW_DOS.COM causes the DOS command window to appear several times.

No action is required from you until the following prompt is shown:

6. "Do you want to install the Define Server Logon and Define AS/400 Information Icons?"

Respond YES.
7. "WIN32S is installed after the run-time installation. Re-boot after the WINS32S installation."

Respond OK.
8. After you receive a message about needing to re-boot in case new paths have been added to Autoexec.bat, press Exit to finish the installation.
9. Your Windows desktop has two new folders:
 - a. Software Installer that contains the Installation Utility
 - b. VRPG run time that contains:
 - Define Server Logon
 - Define AS/400 Information

10. You should not need to re-boot; merely close the Windows session and restart it.

General Considerations for Windows 3.1

Running VRPG Client applications in a Windows 3.1 environment requires the programmer to watch for certain environment specific behavior. This section contains a summary of these Windows 3.1 specific features.

Application Icons

Occasionally, after the VRPG run time or VRPG applications are installed in Windows, the correct icon may not appear in the program group. To create the icon, select the item in the program group, choose File, and then Properties from the Program Manager menu bar. When the dialog appears, press the **Change** push button. This updates the icon.

Project File Names

When creating a VRPG application for Windows, avoid using special characters such as accented characters for the **Source file name**. Some of these characters are changed when the application is installed on a Windows system, which causes the file names to be invalid.

Improving Performance

The performance of VRPG applications can be significantly improved by using a disk caching program such as **SmartDrive**. A disk caching program uses a portion of memory to simulate a disk drive.

Windows 3.1 for WorkGroups

To run Client Access for Windows with Microsoft Windows 3.11 for Workgroups, you must install the Microsoft Data Link Control protocol software. This software is free and can be obtained from Microsoft's Internet home page. You also have to increase the default settings of the protocol (SAPs=>8, xsaps=>5, and xstations=>5). For details, see *Inside Client Access/400 for Windows 3.1*, GG24-4429-00.

Printing

On Windows 3.1, the printer output generated by VRPG applications does not go through Windows print manager to be printed. It goes to the port associated with the printer queue.

Restrictions

Number of parts

Windows has a restriction on the number of window parts that can exist at one time. If you have an application that creates many parts, you may experience run-time errors.

To minimize the number of parts that are created, consider using the SHOWWIN and CLSWIN operation codes, which create and remove windows in memory, if necessary.

Message Help

Second-level help for messages cannot be displayed in Windows. If a message box is displayed with a message that has second-level help defined, the **Help** push button is not on the message box.

Parts, Events, Attributes

Refer to the Chapter 12, "GUI Designer Notes" on page 85 for details of specific Parts, Events, and Attributes that may differ in the Windows run time.

Note:

- DDE Client is not supported.
- Drag and drop is not supported.

Other Notes

Length of names

If you create VRPG Client projects on an **HPFS** drive, ensure that the source file names and directory names are no longer than eight characters if you plan to run these applications on Windows or WIN-OS/2.

Timer Applications named TIMER do not run in Windows.

Logon In certain situations, the VRPG Client Logon prompt may require you to press the **OK** button twice. There is a PTF for this.

Extra events

When an action subroutine is handling an event, it is possible for the user to generate more events. In Windows, these events are not queued and are processed immediately by the VRPG run time.

In some cases, this results in run-time errors. For example, if an action subroutine is issuing IO operations to a file, and another event is generated that also attempts IO operations, **host communications** errors can result since the first IO operation was not complete when the next IO was requested. This situation can also occur if, during a **Create** event, an IO operation is being performed, such as filling a subfile, and the user presses a push button on a window that causes the corresponding action subroutine to issue IO requests.

To prevent these errors, you should disable, using the **Enabled** attribute, those parts that issue IO requests until any current IO is complete.

Memory Considerations for Windows

In order to run VRPG applications under Windows on machines with limited memory (RAM), follow this checklist:

1. In the SYSTEM.INI file (in the Windows directory) ensure that **PageOverCommit** in the **[386Enh]** section is set to at least 10.

The default value for PageOverCommit is three and its range is from zero to 20. This value is used to determine how much disk space Windows allows you to set aside for Virtual memory relative to the real memory installed in your machine:

Allowable_size = PageOverCommit_value X Real_memory

If the **PageOverCommit** statement is not already in the SYSTEM.INI file, it should be placed in the [386Enh] section.

2. If you are in Windows, shut down Windows and restart it; otherwise, just start Windows.
3. Open the Control Panel found in the Main program group in Program Manager.
4. Open **386 Enhanced** and select the **Virtual Memory** push button.
5. A dialog is displayed detailing the swap file currently being used for Virtual memory. The size given is the number that may need to be changed to allow larger programs to run on machines with limited memory. When **Change** is selected from this dialog, the dialog expands and it is possible to change the type and size of swapfile. ADTS CS is unaffected by the type (permanent or temporary); the selection is left to the user.

The size to use depends on the user's configuration and the size of the application being run. For instance, in order to run a simple host connected application such as the provided Subfile sample application on a machine with 8MB installed, the Virtual memory size required to run the application is approximately 36MB (or 36000KB) depending on what other programs are running concurrently. On a 12MB machine, the Virtual memory size only needs to be 32MB.

To run a simple, non-host connected application such as the Resize or Listbox samples, the Virtual memory size only has to be 14MB, which is probably smaller than the default Virtual memory size on an 8MB machine, which means that for such a program, no modifications to the SYSTEM.INI file or to the Virtual memory size are required by the user.

Since the default value for PageOverCommit is three, and $3 \times 8\text{MB} = 24\text{MB}$: On an 8MB machine, you must add a **PageOverCommit** statement to your SYSTEM.INI file to run a host connected application such as Subfile, and set PageOverCommit to at least five so that Windows allows you to specify a large enough Virtual memory size.

Windows has one more restriction on Virtual memory size that users should be aware of: it can be at most half the size of the available disk space on the indicated drive.

For instance, if drive E: is specified as the drive to be used for Virtual memory, and it has 70MB of free space, regardless of what value PageOverCommit is set to, and regardless of the amount of RAM in the machine, the largest Virtual memory size that Windows allows is 35MB.

Client Access for Windows Considerations

The following sections contain some information about Client Access in Windows. Client Access is a prerequisite when using the VRPG run time to connect to the AS/400 system.

Code Page

For Windows 3.1 run time, character code page translation is based on the Client Access setup (using the AS/400 INZPCS command) and the PC keyboard code page. You should ensure that the environments are set up correctly for proper data translation between your workstation and the host.

Network Options

When running applications that involve communication with the host, Client Access for Windows must be installed with network options.

PTF SF28451

To ensure that VRPG Client applications run properly in the Windows environment, Client Access PTF SF28451 for 5763-XC1 must be installed. Without this PTF, applications may hang when run a second time.

Startup

Start Client Access for Windows before starting the VRPG application.

QUSER Profile

Client Access for Windows requires an active QUSER user profile.

Most sites have disabled this profile for security reasons.

Using Windows 3.11 for WorkGroups

To run Client Access for Windows with Microsoft Windows 3.11 for Workgroups, you must install the Microsoft Data Link Control protocol software. This software is free and can be obtained from Microsoft's Internet home page. You also have to increase the default settings of the protocol (SAPs= 8, xsaps0= 5, and xstations0= 5). For details, see *Inside Client Access/400 for Windows 3.1*, GG24-4429-00.

Configuring DB2 for Windows Access

The following sections contain information pertaining to VRPG Client Windows applications that access DB2.

IPF_PATH Conflict

VRPG Client applications cannot run on Windows 3.1 if DB2 Windows Client V2.1 is installed on the same machine unless the environment variable IPF_PATH is removed or remarked (REM) out of the AUTOEXEC.BAT file.

DB2/2 FixPak

If your Windows client application connects to more than one DB2 database from a Windows client running Version 2 of DB2 that require different USERID/passwords, you may receive an SQLCOD equal to -1093 (user is not logged on) instead of a -1403 (user name/password supplied is incorrect).

To correct this, do one of the following steps:

1. Install DB2 Version 2 Windows Fixpak 8081 for CAE and SDK:

You must still check the sqlcod for -1403 and connect a second time.

For example:

```
c/exec sql connect to ctdbt6v in share mode
c/end-exec
c                if                sqlcod = -1403
c/exec sql connect to ctdbt6v in share mode
c/end-exec
c                endif
```

The second connect prompts you to enter a new USERID/password.

2. Explicitly log on with the new USERID/password before starting your second application.

Open the DB2 Group, and double-click on the LOGON icon.

3. You can log on or log off from within your VRPG program before doing your CONNECT (the names of the EXEs provided are only valid on a Version 2.1 DB2 Windows client:

```
D logoff          PR          CLTPGM('db2logff')
D logon           PR          CLTPGM('db2logon')

C                callp        logon
```

4. If you are building your application on an OS/2 system that has Version 2 of DB2, you can use the user/using on the connect statement:

```
c/exec sql connect to dbon40z user latona using mypassw
c/end-exec
```

Note that the Version 1 DB2 client does not prompt you to enter a password.

You must explicitly log on, or use implicit connections.

Connect Reset

Ensure that a **CONNECT RESET** is performed before exiting your application in Windows 3.1. In some situations if the application does not do a **CONNECT RESET** before exiting the application, the Windows kernel may not clean up correctly.

Chapter 6. Win-OS2 Run Time

The environment is almost identical to the Windows run-time environment, so the preceding chapter should be read and understood first, and can be consulted for the appearance and details of the dialogs.

For specific restrictions and variations related to parts, events, and attributes, refer to Chapter 12, "GUI Designer Notes" on page 85.

General Notes

You must be running an OS/2 Warp version (Warp for Windows, Warp Connect, or Warp Version 4 (Merlin)) to execute VRPG applications under Win-OS2.

You **cannot** use OS/2 2.1 or 2.1.1.

Warp for Windows: Win32S

Please refer to "Warp for Windows: Win32S" on page 51 for special instructions for setting up this environment.

Installing WinOS2 Run-Time Support

This is practically identical to the procedure for Windows.

Each workstation that is used to run installed VRPG applications under Win-OS/2 requires this procedure, even if it has the VRPG development platform installed.

The procedure needs to be done only once per workstation.

1. Open WINOS2 Full Screen.
2. Run **Install.exe** from the directory into which you packaged the Windows run time.
3. Press **Continue** on the install prompt.
4. When prompted, type the directory into which the Window run-time support code should be installed.

This can be a LAN drive.
5. EPFW_DOS.COM causes the DOS command window to appear several times.

No action is required from you until the following prompt is shown:

6. "Do you want to install the Define Server Logon and Define AS/400 Information Icons"?

Reply YES (this makes run-time administration **much** easier later!)
7. "WIN32S will be installed after the Run-time install. Re-boot after the WINS32S install".

Respond OK.
8. After you receive a message about needing to re-boot in case new paths have been added to Autoexec.bat, press Exit to finish the installation.
9. Your Windows desktop has two new folders:

- a. Software Installer that contains the Installation Utility
 - b. VRPG run time that contains:
 - Define Server Logon
 - Define AS/400 Information
10. You should not need to re-boot; merely close the WIN-OS/2 session and restart it.

Win32S Run-Time Support: Post-Install Checks and Note

1. If you have an old version of WIN32S, this is the time to detect and fix the problem.
 - Click on the Define Server logon.
 - If you get the Win32S error "Missing numeric co-processor virtual device driver [VMCPD]", you need to get the WIN32S 1.25a (or later) support.

To run VRPG Client programs under IBM OS/2 WARP Version 3.0, you need the latest version of the WIN-OS/2 support for WIN32S 1.25a. This version was available only in BETA when VRPG R6 was first generally available. It now is available in an official OS/2 Fixpak.

You can get this code at any of the following places:

The Readme suggests the following method, which is no longer valid:

Outdated Information from Readme File

Do not use.

- In section 12 of the PSPBETA FORUM on CompuServe, use GO PSPBETA.
- On the IBM TalkLink OS2BBS.

Fixpak 26 for OS/2 WARP V3 is now generally available. It contains WIN32s level 1.25b support.

Information about the Latest Fixpak's Available

The following URL has information and links to the download files:

http://ps.software.ibm.com/fixnews.html#Header_2

Also OS/2 Warp V4 (Merlin) contains the 1.25b level WIN32s support. If you have WARP V4, there is no need to download additional WIN32S files.

2. If you are installing VRPG Client on a DBCS system, refer to "DBCS Considerations" on page 83 for information on installing WIN32S on a DBCS system.

3. Set your WIN-OS/2 Compatibility Mode:

To ensure that Windows 3.1 programs generated by VRPG Client run seamlessly on the OS/2 desktop, WIN-OS/2 must be set to run in Enhanced mode. To set the mode, use the following steps:

- a. Open the WIN-OS/2 Setup settings notebook in the System Setup folder.
- b. Select WIN-OS/2 settings.
- c. Press the OK push button.
- d. Ensure that WIN_RUN_MODE is set to 3.1 Enhanced Compatibility and press Save.

Note: Sometimes the WIN-OS/2 settings notebook shows that the current setting is 3.1 Enhanced Compatibility, but WIN-OS/2 still runs in Standard mode. To get around this problem, set the WIN_RUN_MODE to 3.1 Standard, select Save, close the settings notebook, and repeat steps 1 through 4.

4. Start the WIN-OS/2 server:

VRPG Client Windows applications running in the WIN-OS/2 environment that communicate with an AS/400 system require a special server program, FVDCSERV.EXE, to be running.

You can start the server by entering FVDCSERV.EXE on an OS/2 command line or by placing it in your startup folder. Once the server is started, it remains active in memory and does not need to be restarted unless you shut down OS/2.

The server can be removed from memory by running the program again with the termination parameter:

```
FVDCSERV /d
```

5. Notes on parts and attributes:

The following describes differences in the behavior of parts and attributes between the OS/2 and WIN-OS/2 run-time environments:

- Due to the differences in font sizes between Windows and WIN-OS/2, the cursor may not always appear in entry fields. Data can still be typed in the entry field. To circumvent this problem in WIN-OS/2, increase the height of the affected entry field, or choose a smaller font.
- For the multi-line edit part, the Wordwrap attribute does not function if the horizontal scroll style is set.
- The HelpEnable attribute for the push button and graphic push button part does not work in the WIN-OS/2 or Windows run-time environments. To display the help, give the part focus and press F1.

6. Other notes:

- When running Warp for Windows, if you start a VRPG Windows program and then start a VRPG OS/2 program and subsequently terminate the Windows program, the windows of the OS/2 program are minimized. You can restore the windows using the window list.

WinOS2 Run Time - Settings Notebook

You can change the run-time AS/400 information using this notebook.

Refer to the OS/2 material in the previous chapter. The appearance and function is nearly identical.

Chapter 7. Pre-Installation Checks

This chapter contains tips on how your workstation should be set up before you install ADTS/CS. Adhering to these recommendations avoids surprises during installation and usage of ADTS/CS.

AS/400 Environment

1. Make sure that ADTS is installed on your AS/400 system. ADTS contains PDM, SEU, SDA, and so on. If you use these products, ADTS is installed on your system.
2. Check that the latest CUM tape has been applied, or at least that all available relevant PTFs are installed.

Note: Latest-PTF List

Keep checking the VRPG Home Page at <http://www.software.ibm.com/ad/varpg> for the latest fixes.

The following PTFs *must* be applied:

- a. PTF SF20554 (5763-SS1 Communication fixes)
 - b. For product 5763-PW1 (ADTS)
 - PTF SF25794 for V3R1
 - PTF SF67924 for V3R6
3. For each programmer that utilizes the ADTS CS environment:
 - a. Check that the user name and password are eight characters or less.
 - b. Create an individual EVFCICFF file on the development AS/400 system.

Installing Application Development ToolSet Client Server for OS/400 Version 3, SC09-2188, has more information on the ICF file that is needed. Also check "Dealing with Different Network IDs" on page 16 if you are in a network environment that deals with multiple NETWORK IDs.

Ensure that the library containing this file is in the normal library list of the programmer.

If it is not, one of two things may happen:

- The STRCODE that connects the development platform to the host may fail.
- Another person's EVFCICFF file may be used and, therefore, become locked so they cannot use it.

OS/2 and Communications Layers

Note: OS/2 Warp is really the requirement; there are restrictions using 2.1. Warp Connect is preferable.

OS/2 2.1

Apply the fixes contained in the OS/2 Version 2.1 ServicePak (also known as OS/2 Version 2.1.1).

- If you do not want to install the OS/2 Version 2.1 ServicePak, a patch is available that fixes a shut-down problem with OS/2 and several problems with WorkPlace Shell. This fix is available online from several networks, and is distributed as 21WPSF.ZIP. The fix can also be obtained by contacting the IBM support center and asking for a fix to APAR PJ08727.

If you do not have this fix installed, please ensure you allow the shutdown to complete fully, especially during installation of the WorkFrame/2 product. This may take several minutes to complete.

In North America, you can dial 1-800-237-5511 to contact the support center.

- If you are using OS/2J (Japanese version), you must install the fix to OS/2 APAR PJ13793 before installing WorkFrame/2. Contact the IBM support center to obtain this fix.

OS/2 Warp

You are advised to use Warp Connect, as it presents significant advances in communications configuration.

LAPS

If you are *not* using OS/2 V3 Warp Connect or Warp Version 4 (Merlin), you need to install and configure the LAPS (LAN Adapter and Protocol Support) package to implement NTS/2 (Network Transport Services).

Warp Connect: MPTS

Warp Connect and Merlin provide MPTS (Multi-Protocol Transport Services), which contains the LAPS layer.

Use MPTS Configuration to add the 802.2 protocol under the LAN Adapter and Protocol Support (SNA only).

Communications

The ADTS CS development environment requires support for CPI-C and 5250 Emulation.

The VRPG application run-time environment requires support for CPI-C. CPI-C is a high level set of APIs that uses SNA APPC (LU6.2) communications protocol.

Install and configure one of the following products.

CM/2 (Communications Manager/2)

CM/2 level 1.1.1 is the preferred product at the time of this residency.

Prior levels of Communications Manager *might* work, but have not necessarily been tested.

CM/400

Client Access/400 Communications Manager does not include 3270 support, connection network support, and Synchronous, Asynchronous, X.25, and ISDN connection types. However, it simplifies some aspects of configuration, and has a better interface to Client Access/400 Optimized for OS/2.

Communications Server and OS/2 Client

We have no experience with this latest set of communications products. ADTS CS has not been tested against them in the Lab. However, field reports indicate this product set is OK.

Personal Communications AS/400

We have not tested this. ADTS CS has not been tested against it in the Lab.

Note

Whichever Communications middleware is selected, check that the LU alias is eight characters or less.

Client Access

The development platform requires Shared Folder support to access product code and fixes on the AS/400 system.

Run-time clients *may* require PCS or Client Access to be installed, for instance, if they call CA APIs, executables, or commands such as RUNRMTCMD. You need to know the application requirements to determine this. If in doubt, install Client Access/400.

Install and configure one of the following products:

1. PC Support
2. Client Access

CA/400 with Warp Connect

If you plan to use Client Access/400 running on Warp Connect, you must install Warp Connect, any Warp Connect Networking Support you plan to install, and the ExtendPak for Warp Connect (if applicable) first.

You may then install Client Access/400.

After installing Client Access/400, you must go into the MPTS configuration and add the 802.2 protocol under the LAN adapter and protocol support (SNA only).

Note that if you install Warp Connect after Client Access/400 has been installed, you must reinstall Client Access/400.

CA/400 Install over NetWare

When installing CA/400 over NetWare, Part 2 of the installation will fail with a SYS2070 message:

System cannot demand load app segment, MUG->NETAPI.80 is in error.

The problem is caused by Communications Manager and NetWare both using NETAPI.DLL. To prevent this problem, use the following steps:

1. Reverse the order of the NETWARE and MUGLIB entries in the LIBPATH statement of the CONFIG.SYS file.
2. Restart the PC.
3. Start the Client Access/400 installation part 2 again.

After installing Client Access/400, the NetWare configuration may need to be changed. Client Access/400 requires NDIS, so the changes may include switching to the NTS/2 NetWare support which maps ODI over NDIS.

Co-Existing Products

1. Lotus SmartSuite:

Install this *before* VRPG Client. Otherwise, the EPFIS.INI (Program Information) file in the OS2\SYSTEM directory used by VRPG becomes corrupted, and you may have to re-install VRPG.

2. Visual Age C + + V3.0 for OS/2:

Install this *before* VRPG Client. The C++ product and VRPG share certain components and the VRPG versions are more current.

3. Install OS/2 MMPM/2 Multimedia extensions:

If the VRPG Audio Part is to be used in the OS/2 environment, MMPM/2 must be installed.

4. Ensure DOS Level is at least 6.3 if you have a DBCS system and are running VRPG Windows applications.

Just Before You Start Installing

1. Make sure no VRPG, CODE, LPEX, or WorkFrame componentry is active.
2. Make sure that TCP/IP for DOS is stopped before installing VRPG run time on a Windows System.

3. If you have a previous release of VRPG run time, delete it.

Otherwise, error SYS2070 occurs during the second phase of the installation.

To delete the old run time:

- a. Make sure all VRPG applications and tools are closed.
- b. Use the following command to delete the run time:

```
d:\path\VRRINST /a:d
```

where *d:\path* is the directory containing the VRPG run time.

Chapter 8. Installation Problems and Errors

In case you run into problems during installation, make sure you read this chapter. It contains error scenarios and tips on how to recover.

If WorkFrame/2 Phase 2 Failed

If the WorkFrame/2 installation failed, you need to retry the phase 2 installation of VRPG Client/2.

The most common error is that WorkFrame/2 classes cannot be registered. If this happens, you see a message such as:

Error - Failed registration - 'DDE3xxxxx'

The following steps describe how to retry Phase 2:

1. Complete the installation of the product.
2. Open the Startup folder in the OS/2 System folder. You see the "Create Code/400 Projects" program icon.
3. Delete the icon from the startup folder. Either use the right mouse button and the Delete option, or drag the icon onto the shredder.
4. Go to an OS/2 window and enter the following command:

```
FVDEUN C E:\IBMWF
```

where *C* is your boot drive and *E:\IBMWF* is the directory where WorkFrame/2 is installed.

5. Shut down and re-boot your system.
6. After re-booting, wait until all activity on your workstation is complete.
7. Open an OS/2 window and enter the following command:

```
EVFICPAM C 400 INSTALL WFDESK
```

where *C* is your boot drive.

8. You see the "Create CODE/400 Project" window that de-registers and registers the WorkFrame/2 classes and also creates some WorkFrame/2 sample projects and action profiles.
9. This is followed by the Communications Manager/2 progress indicator and the Communications Manager/2 completion message.
10. Wait until you see a "Create CODE/400 Projects: message that says:
This session may contain an active program Do you want to close this session without saving data?

You can select either Yes or No to close the window.
11. This is the end of Phase 2.

If Communications Manager/2 Failed

1. If you get a Communications Manager/2 error, check the C:\CMLIB\CMRINST.LOG file and correct the mistake.

On an OS/2 window, execute the following command to repeat the Communications Manager/2 update:

```
TPUPDATE
```

2. If you get an error from Communications Manager/2 that says:

INS2250. The source path specified in CMSource is not valid. Check that the directory specified exists and can be accessed.

This may be because you installed Communications Manager/2 from LAN and the LAN drive is no longer attached to your workstation. Re-attach to the LAN drive first and use the TPUPDATE command.

3. If your CMLIB\CMRINST.LOG says you have an invalid transaction profile because it is pointing to the wrong directory, it usually means that you have previously installed Code/400 and your CONFIG.SYS still points to the old directories. Remove those entries from the CONFIG.SYS file; re-boot and use the TPUPDATE command to repeat the Communications Manager/2 update.

System Errors

1. If you get a system error from CODESERV.EXE that says:

The system could not demand load the application's segment.
EVFCBASE->EVFUTL32.142 is in error.
end this task.

This is usually caused by an old level of LPEX. This can happen if you have other products that also have LPEX shipped with them. Remove their entries from the CONFIG.SYS file and re-boot your system.

2. If you get an error saying the VRPG sample application and the action profile cannot be deleted, you have probably performed an Undo Installation from the WorkFrame/2 Setup.

This de-registered the classes that the VRPG objects are based on and, therefore, cannot be deleted using WorkPlace Shell. The only way around this problem is to go the file system and manually delete the directory represented by the ADTS Client Server/400 folder along with any files inside it. This directory can be found in the desktop directory of your boot drive. It is probably named either "ADTS_CLI" for FAT systems or "ADTS Client Server!400" for HPFS systems.

3. If, during the installation of LPEX, you get an error indicating the file LPEX.PKG is not found, cancel the installation and install LPEX again.
4. During Phase 2 of the VRPG Client/2 installation, **do not shut down or re-boot your workstation** until you see a message with the heading **Create CODE/400 Projects** that says: "This session may contain an active program. Do you want to close the session without saving your data?"

This indicates the end of Phase 2 of the installation. You can select either Yes or No.

5. If you want to delete the VRPG Client/2 product, do not de-install WorkFrame/2 from the WorkFrame/2 Setup window first. If you do, the **CODEINST** delete action cannot remove the sample applications and the action profile later. You need to manually remove them by going to the desktop subdirectory.
6. After you have installed VRPG, if you need to install another product that also contains SOM Dynamic Link Libraries (DLLs), check the date of the SOM DLLs used by that product. If they have later dates than the SOM DLLs used by VRPG and they are compatible with older SOM DLLs, the directory containing their SOM DLLs should be placed in front of the VRPG SOM directory in the LIBPATH of your CONFIG.SYS file.
7. If you have CODE/400 already installed on a workstation, install VRPG Client/2 in the same directory.

8. If you have WorkFrame/2 already installed on a workstation, install the new WorkFrame/2 using the VRPG Client/2 installation. The new WorkFrame/2 must be installed in the same directory as the current WorkFrame/2.
9. If error EPFIE234 occurs, the only known solution is to re-install OS/2 Warp (without reformatting the boot drive) and then re-install the ADTS CS products.

Error Logging

A file named **FVDCERRS.LOG** is created in the CODE directory for development projects and in the project run-time directory (the default is VRPGRT) for installed applications.

If you are experiencing communication problems during build time or run time, you can use an editor, such as E.EXE or EPM, to view this log for possible problems. The file created in the source directory contains logging information generated when the program was being compiled. The file created in the run-time directory contains logging information generated when the application was being executed.

Note: You may get a message indicating the file is locked. If that happens, use the following command to view the file:

TYPE FVDCERRS.LOG | MORE

Chapter 9. After You Install

Remember that to begin working with existing VRPG projects, it is necessary to re-boot the system after Phase 2 of the installation is complete.

PTFs

After the initial installation is complete, the latest PTFs related to the ADTS CS products need to be installed.

If you install from the AS/400 folders, the first attempt to use the STRCODE function (explicitly or implicitly) causes the product PTF levels on the workstation and host to be compared. If the host is at a later fix level, the workstation user is prompted to install the fixes. The installation can be bypassed, but we recommend that you do accept the updates as soon as they become available.

A list of the latest PTFs is maintained at the VRPG home page at <http://www.software.ibm.com/ad/varpg/>.

The file EVFCCL62.LOG records the client PTF application activities.

Warp for Windows: Win32S

If you have installed VRPG Client on Warp for Windows, the **WIN32S** support required to run VRPG Windows applications from OS/2 is not installed. You must install this support using one of the following methods. Which method you use depends on where you have installed VRPG Client. If you are installing VRPG Client on a **DBCS** system, refer to "DBCS Considerations" on page 83 for information on installing WIN32S on a DBCS system.

1. VRPG Client installed on an HPFS drive:

If you have installed VRPG Client on an HPFS drive, use the following steps:

- a. Use **XCOPY** to copy the WIN32S installation files to a drive accessible to Windows. These installation files are shipped with VRPG Client. Assuming VRPG Client is installed in **H:\CODE**, the following command copies the installation files to a directory named TEMP on the C drive:

```
XCOPY H:\CODE\WIN32S\DISK1 C:\TEMP\
```

Note: Approximately 1.6MB of disk space is required to copy these files.

- b. Boot the system to DOS and start Windows.
- c. From the Windows Program Manager, select **File -> Run**.
- d. In the dialog, enter **C:\TEMP\WINSETUP** on the command line prompt and press **OK** to install WIN32S.
- e. Once the WIN32S installation is complete, you may delete the TEMP directory.

2. VRPG Client installed on a FAT drive:

If you have installed VRPG Client on a FAT drive, use the following steps:

- a. Boot the system to DOS and start Windows.
- b. From the Windows Program Manager, select **File -> Run**.
- c. In the dialog, enter **C:\CODE\WIN32S\DISK1\WINSETUP** (assuming VRPG is installed in C:\CODE) on the command line prompt and press **OK** to install WIN32S.

Whichever method you choose, you must modify the AUTOEXEC.BAT file in the OS/2 boot drive by adding WINTEST and WINDLL to the beginning of the PATH statement.

For example, if you installed VRPG Client in C:\CODE, the PATH statement should look similar to this:

```
PATH C:\CODE\WINTEST;C:\CODE\WINDLL;....
```

Also, you must add the following statement to the AUTOEXEC.BAT file in the OS/2 boot drive (this example assumes you have installed VRPG in C:\CODE):

```
SET HELP=C:\CODE\HELP;C:\CODE\WINTEST
```

Clean Up Old Debug Profiles

After installing the new version of VRPG Client, remove all old debug profile information from your system that may have been created by the earlier version of the debugger.

The debugger creates the profile in files named "name.@CR" and "ICATVRPG.@CS" where name is your project name. The easiest way to clean up old profile information is to do a scan for the file "*.@" and type the command "DEL *.@" . These files are usually located in the OS2 directory of the boot drive.

Chapter 10. Some Errors That May Be Encountered

This chapter contains information about some of the errors you may encounter during the installation and usage of ADTS/CS.

This list provides a fast cross reference to certain error conditions:

- "Before You Start"
- ALT2438: ALERT Local SNA Unable to Establish Session on page 56
- EPFIE205: Project Already Installed on page 56
- EPFIE234 on page 57
- FVDE36653: Query Cannot Be Performed on page 57
- INS2250: Source Path Specified in CMSource is not valid on page 57
- 0202 The call to program xxxxxxxx ended in error on page 57
- 0940 Error occurred in host services on page 58
- 0940 Error occurred in host services on page 58
- -E RC:error - Unable to find file on page 59
- Error - Failed registration - DDE3xxxxx on page 59
- EVFCBASE- EVFCUTL32.142 in Error on page 59
- Failed to establish communication session on page 59
- Missing numeric co-processor virtual device drive on page 59
- Return code from SAVE was -6 on page 60
- The communications link to the server is broken on page 60
- The file or user space xxxxxxxx, or both, cannot be used on page 61
- The specified server alias name is not a simple name on page 61
- The system could not demand load the application segment on page 61
- VRPG Client - Build (OS/2). xxxxxxxx.DLL is locked on page 62
- Server name does not show in Define Servers dialog on page 62
- Define Server Logon has no effect on page 62
- Actions are missing from a VRPG project pop-up menu on page 62
- Some Actions on the VRPG Master Project do not work on page 63
- Actions fail after VRPG Create Empty on page 63
- WorkFrame no longer functions on page 63
- Debugger freezes on page 63
- Insufficient Stack Size error on page 63
- "The AS/400 Service Knowledge Base" on page 63
- "RPG Compile and Run-Time Errors" on page 64
- "Table of Run-Time Error Codes" on page 65

Before You Start

Check this information before you get too far into trouble shooting mode. Our experience shows that most problems can be fixed by defining these values correctly.

On the Client Side

As you see when you continue, most of the problems experienced with VRPG configuration are due to mismatching the LU name, LU alias, and Partner LU alias between Communications Manager and VRPG.

If you are not completely familiar with this information, you need to collect and understand it.

You need to find the following Communications Manager data:

Note:

Characters depicted in a reverse image block such as **E** refer to the communications variable in Table 1 on page 4.

1. LU name of your workstation **E**
 2. LU alias of your workstation **G**
 3. PLU alias of your AS/400 system **H**
- Check these against the VRPG definitions in the [Project] [Define AS/400 Information] [Servers] tab. The host PLU alias **H** must appear as the "Remote Location". Getting this correct sends you on the way to running the application.
 - However, working with the GUI Designer requires that you have the STRCODE server parameters correct.

Normally this is not a big deal; VRPG starts the CODE server automatically and fills in the correct parameters for you.

In case this automatic CODE server start up does not work, issue a STRCODE command from a 5250 emulation window.

On the 5250 command line, enter

```
STRCODE <the server name (or PLU alias) H of your host>  
<the CM LU name E of your workstation>.
```

Note:

Make sure that these parameters are entered correctly; otherwise you might have the CODE server running but VRPG does not recognize it. If you use the default in the STRCODE command for "Host server name" which is "OS400" and your AS/400 PLU name **H** is different, the CODE server is started, but the VRPG compiler does not recognize it because of the wrong name and it does not resolve your external descriptions.

- The first parameter (Server, meaning "CODE server" not "Host") is the PLU alias or server name **H** you specify in VRPG components using the [Project][Define servers] dialog. You need to use the same name on STRCODE as you have in your server definition.
- Now if you get the second parameter (Remote Location **E**, meaning the PC) wrong, you might still hear the famous three beeps somewhere around you but not on your workstation! That means you have entered someone else's workstation LU name in parameter 2.

If you use multiple terminals and use STRCODE without parameters, you may discover that STRCODE remembers the parameters by *user* and not workstation!

- Furthermore, if you do not have the EVFCICFF file in your library list, or if you use someone else's EVFCICFF file in the library list before yours, or if you already have a STRCODE running, you can have associated problems.

On the AS/400 System Side

- Make sure the EVFCICFF file is set up correctly in a library contained in the library list of the job description (JOBDD) used for the batch job submission. The EVFCICFF file can also be in the library defined as the current library (CURLIB) in the user profile (see additional information in the *VRPG Client/2 Installation Guide*). Use the DSPUSRPRF command and WRKJOBDD command to determine the job description, current library, and library list being used.

The library list used by the jobs invoked on the host is always of great importance to VRPG. Each user profile, whether for developer or end-user, should reference a library list that is suitable for all of the ADTS CS activities and environments expected. The EVFCICFF file must be in the library list, but to make life easier from here on, the source, data, and run-time libraries should also be in the user's normal library list.

- Some communication sessions in VRPG Client/2 are accessed by submitting a batch job request to the host. Therefore, it is important to check that the setup in the host batch job subsystem does not prevent these remote requests from starting up.

Putting the QBATCH job queue on hold status, limiting the number of active jobs under the QBATCH subsystem, and causing a long delay for the batch job to start up, and so on, may cause the workstation communication request to fail.

Use the WRKACTJOB, DSPSBSD, WRKJOBQ, and CHGSBSD commands to determine if the batch job request from the workstation was started.

- Some communication sessions in the VRPG Client/2 environment are accessed by submitting a batch job to the AS/400 **QINTER** subsystem. Verify that the setup for the QINTER subsystem and the QINTER job queue in the QGPL library do not prevent these remote requests from starting up.

For example, if the QINTER job queue is set to *hold* status, or if the number of active jobs allowed in QINTER is exceeded and causes a delay in starting the server job, the workstation communication request may fail.

Use the WRKACTJOB, DSPSBSD, WRKJOBQ, and CHGSBSD commands to determine if the remote request from the workstation was processed.

- Verify that the subsystems QCMN and QINTER are active.
- Use the WRKSBMJOB command and WRKSPLF command to determine the status and result of the workstation job request.
- If AS/400 VRPG jobs get terminated after they are idle for some time, you might want to check the AS/400 system value QINACTITV (Job Inactivity Timeout). If this is not *NONE, the system is terminating these jobs after the scheduled inactivity time. Set this system value to *NONE to avoid surprises with your VRPG or CODE/400 server jobs. The default for this system value is *NONE. From our experience, you only run into an inactivity value being set on secure systems.

VRPG Client/2 can only communicate with a host that has met the VRPG Client/2 installation requirements.

Foremost of these is that the host be *at least* at V3R1 level.

Error Messages with Message IDs

These come from any and all of the components we have to deal with. They are in message ID sequence.

ALT2438 ALERT Local SNA Unable to Establish Session

(This is caused by an internal error).

Attempting to start CM/2 communications followed by "The logon did not complete successfully".

This might be telling us that we need to re-install CM/2, but there is a chance we can recover from it by:

1. Shutting down communications (abnormally, if necessary).
2. Getting onto an AS/400 command line elsewhere.
3. Entering WRKCFGSTS *CTL myLuname **E**.
4. Varying OFF the controller and device, which might be showing Recovery Pending, or Vary On Pending.
5. Varying ON the controller and device.
6. Starting communications again.

EPFIE205 Project Already Installed.

(You cannot install a product twice).

You are running VRAINST.EXE directly from a packaged application directory.

The Installation process, whether run directly as VRAINST.EXE or by using the Installation Utility, updates the local list of installed applications. If you are going to re-install your application, you have to persuade the Installation system that either this is a new version, or you have deleted the old version.

You can use the installation utility to delete (its record of) the previous version of your product. If you do that and your package and installation directories are identical, you lose your .PKG file and, therefore, are prevented from re-installing (you must re-package).

If you use a version numbering (of the four identification numbers provided in the Packaging utility, only version is used by the Installation Utility to separate instances of the same package name), and the current installation has a different version number than the previous (or any) installed version, you do not see this error. However, if you have installed the later version or versions into the same target directory, and you use the Installation Utility [Delete] function, you have, nonetheless, deleted your current application.

By the way, VRAINST.EXE does not work if the Installation Utility is open at the same time; it just returns to the command line.

What To Do This Time

1. Copy your current packaged application to some safe place.
2. Use the Installation Utility to delete the current product's registration:
 - Run the Installation Utility.
 - Take [View][Installed Products] so you can see your application.
 - Select your application.

- Take [Actions][Delete] to remove your product. Remove old instances also.
 - Close the Installation Utility. VRAINST does not work if it is still open.
3. Copy your packaged objects back to the installation directory.
 4. Run VRAINST.

What To Do in Future

1. Use *different* directories for packaging and installation. You can use a common packaging target directory, but have separate directories for the installed components.
2. Use version numbering. Understand what happens when you delete old versions using the Installation Utility.

However, if you follow these instructions, your packaged objects are still intact, so you can just delete any old versions you want. Then go ahead and install.

EPFIE234 This error occurs during Phase 2 of the ADTS CS installation.

This is noted in the V3R1 CODE/400 ReadMe First. We have not encountered it.

The only known cure is to re-boot (without reformatting the boot drive).

EPF1077W

Host Server Busy or Not Available

This error occurs when a second CODEEDIT is run before the previous CODEEDIT has finished starting. The parser does not get started in this second window.

The known cures for this release are:

1. Run multiple servers.
2. Do not do it! Wait for the first session to start properly.

FVDE36653

Query Cannot Be Performed

This has been experienced when there are *shadows* of network folders on a desktop, which display broken links until they are "touched". Early on, at least, trying any WorkFrame actions while the links showed broken produced this error message. May have been cured in the current WorkFrame anyway.

INS2250 Source Path Specified in CMSource is not valid...

You installed CM/2 from the LAN, and the LAN drive you need is no longer attached.

Check your *cmlib*ICMRINST.LOG.

Then re-attach the drive, and run TPUPDATE again.

(This is fixed in VRPG and CODE V3R1.1, only occurs in V3R1.)

0202 The call to program xxxxxxxx ended in error

Perhaps your program on the AS/400 system did blow up, but maybe the call never even got to your host program.

1. You might have a library list problem. In the GUI designer, try selecting:

[Project]

[Define AS/400 Information][Programs]

[Programs]

Define the AS/400 program you want to call with an explicit library entry.

Then try your call again.

This is probably only a temporary solution: eventually you want to get your library lists permanently correct.

2. Something does not recognize a PU or LU name. See the suggestions that follow for X'00000001' host services error.

0940 Error occurred in host services

(With reason code X'00000001')

A VRPG run-time error message window: you are trying to run a VRPG application that calls a host program.

The Partner LU Alias **H** you have entered in the VRPG [Define AS/400 Information][Server] does not match the PLU Alias definition in Communications Manager. (It is not the "Server Alias" but the "Remote Location" we are worried about here. The "Server Alias" is an internal reference in VRPG and does not come outside.)

1. Type PMDSPLAY on an OS/2 command line take [Display][GeneralSNA][Partner LU Definitions] to see what CM thinks the PLU Alias really is. You can also get here by using the SubSystem Management icon in Communications Manager.
2. In the VRPG GUI Designer, take [Project][Define AS/400 Information][Servers] and check that the Remote Location cited agrees with the CM PLU Alias.
3. Note that you do not have to do another [Project][Build] to change the server definition for VRPG. You can either just fix the notebook data and select [Project][Save], or manually update the .RST file. Either way, the contents of the .RST file are loaded at run time to determine these things. That applies equally to running from the GUI Editor, selecting Run from the Project folder, or running the installed application from its icon.

Note: If you do not rebuild your application, you have to copy the .RST file into the TEST or WINTTEST directory because on a development workstation, the application is run from these directories.

0940 Error occurred in host services

(With reason code X'0000000F')

A VRPG run-time error message window: you are trying to run a VRPG application that calls a host program.

You have not defined an AS/400 server in the RST file, that is, by making entries in the [Project][Define AS/400 Information].

1. Rebuild the application with a default server defined.
2. You can edit the .RST file directly.
3. Are you using the same server in the run time as you are on the development platform?

Error Messages Without Identifiers

Again, these are from all over, presented here alphabetically ...

-E RC:error

Unable to find file "RCPP.ERR"

This error occurs during the VRPG build. It was reported in the VRPG CFORUM. The probable cause is as follows:

Your path statement exceeds 255 characters and you have not yet installed Fixpak17. Directories in the PATH beyond 255 characters are not included in the PATH search, so these files are not found. They are, in fact, in the code\BIN directory. A temporary fix is to copy them to your project directory.

Error Failed registration - "DDE3xxxxx"

This error occurs during the Workframe installation. WF/2 classes cannot be registered.

Refer to "If WorkFrame/2 Phase 2 Failed" on page 47.

EVFCBASE-

EVFCUTL32.142 in Error

(End this task).

See The specified server alias name is not a simple name on page 61

Failed to establish communication session

(With the workstation).

5250 session message EVF0001 when STRCODE is attempted.

The second parameter of STRCODE is not the LU Alias of your workstation.

Check the job log on the AS/400 system for more detailed messages.

If the message states "Target program not available", run TPUPDATE from the OS/2 command line, stop Communications Manager, start Communications Manager, and try again. TPUPDATE creates the entries for the required ICF programs in the Communications Manager definition.

Also check whether there is already a CODE server job running. In the VRPG product folder, double-click on "Shut down all Compiler servers". If you hear the triple CODE server beeps, you have a CODE server already running.

Missing numeric co-processor virtual device driver

[VMCPD]

Win-OS2 is complaining that you do not have the requisite version of WIN32S run-time support.

To run VRPG Client programs under IBM OS/2 WARP Version 3.0, you need the latest version of the WIN-OS/2 support for WIN32S 1.25a. This version is available in Fixpak 26 for WARP V3 or in WARP V4 (Merlin). Download instructions for Fixpak 26 can be found in

“Win32S Run-Time Support: Post-Install Checks and Note” on page 40.

See also the section on Win-OS2 run-time support (“Warp for Windows: Win32S” on page 51).

Return code from SAVE was -6.

(See Editor reference for details.)

From the VRPG [Project][Save] or SaveAs, or the Editor [File][Save] or SaveAs.

- Saving to an AS/400 system or a LAN drive:

Maybe you do not have a server active. Maybe you have not even *defined* a server.

1. For a LAN drive, check that the drive is still attached.
2. If it is a shared folder, have you started Client Access?
3. If it is CA Optimized for OS/2, have you started the drive by clicking on its icon?
4. Open [Project][Define Servers].
5. If there is no server shown for your SAVE medium, go to an emulator window and do a STRCODE.
6. If *that* fails, start at the beginning of this chapter!
7. Open [Define Servers] again.
8. Select and **Apply** the required server.
9. Select **OK** to return to the project.
10. Try the save again.

- Saving to a local drive:

- Check the ATTRIB of the .VPG file (if you are having trouble saving from the editor) or all the component files (if you are in [Project][Save]. Does it have the READONLY (R) attribute set?

To test the attribute, type the following command on an OS/2 command line:

```
ATTRIB MyName.VPG
or (assuming the required directory is the current directory):
ATTRIB *.*
```

To remove the READONLY status, use this:

```
ATTRIB -r MyName.VPG
or
ATTRIB -r *.*
```

The communications link to the server is broken

(35600 - 37953)

VRPG Builder menu options, such as “Define Reference Fields”

Everything else seems OK?

Look at any messages that were issued behind your STRCODE command. You must look at the lower-level messages in the emulation session (Call QCMD or DSPJOBLOG and use F10).

1. The "Server" name you used on the STRCODE name does not correspond to the Partner LU alias (PLU alias) **H** defined in Communications Manager for the AS/400 system you are using.

The PLU alias is probably on the bottom status bar of your emulation window. If the emulation window does not show the PLU alias, use the OS/2 command PMDSPLY to get information about your current communications configuration (see 0940 Error occurred in host services on page 58).

2. User not enrolled in system distribution directory:

You need to do an ADDDIRE for your profile.

3. User name or password more than eight characters long:

This is a subtle trap. Co-author CONGERTON now answers only to CONGERTN because of the amount of trouble the nine-character name causes at various places in networking, particularly in this scenario.

The file or user space xxxxxxxx, or both, cannot be used

(It is locked by another job.)

During ADTS CS Backup, you receive this message if the project size exceeded the size limit on the physical file used to do the backup. The file should have SIZE(*NOMAX).

The specified server alias name is not a simple name

(41080-41231)

You are trying to define a server in the [Define AS/400 Information] dialog.

The alias you assign has to follow the rules for an AS/400 name. Probably you are trying to use 5250PLU. That is not valid because it starts with a numeric.

Note that this "Server Alias" **J** is purely an internal convenience for your VRPG component. It is, in fact, an alias for the host PLU alias, which is entered as the "Remote Location" **H** in this dialog. You can use any name you want because it is not referenced outside the VRPG component, but it must follow the naming convention.

The system could not demand load the application's segment

Generically, this means the DLLs used by something are of mixed versions.

Something you have installed is seeing an older version of some DLL it uses because the PATH is unsuitable. Or vice versa, an older EXE now sees a newer DLL it cannot deal with.

A specific case:

EVFCBASE- EVFCUTL32.142 is in error. End this task.

This is issued by CODESRV.EXE. It is caused by an old LPEX level. If you have an older product that uses LPEX, you may have problems unless you discontinue using it and remove its entries from CONFIG.SYS.

VRPG Client - Build (OS/2). xxxxxxxx.DLL is locked

(It cannot be updated - please unlock it and retry. (39500-39498))

You execute the application, probably from the builder, and it has not been fully terminated. Shut it down. If you cannot find it, then sorry, you have to re-boot; later check that you have marked the component as one to be shown on the windows list. In V3R1.1, this is the default for the first Window designed for an application or component. The Window title shows up in the OS/2 Window List; just right-mouse click on it and select close.

After you have unlocked the DLL, press the RETRY button.

Other Error Situations

Server name does not show in Define Servers dialog

If you use the **STRCODE** command to start the Code session from an AS/400 emulation session, the server name specified on this command must be a valid PLU Alias Name **H** as defined in your Communication Manager setup on the workstation. If it is not, VRPG does not recognize it as a valid server name, and it does not appear in the **Define Servers** dialog.

An easy way to check the PLU Alias on CM/2 emulation windows (PC5250 does not have it) is to look in the exact center of the bottom status bar.

Otherwise, you should check in the Subsystem Management folder under Communications Manager/2. Follow this trail: [SNA Subsystem] [Display Active Configuration] [Display] [General SNA][Partner LU Definitions]. The Partner LU Alias you see here is the one that is expected throughout ADTS CS.

Define Server Logon has no effect

After the VRPG communication session has started with the host, if you change the logon information using the Define Server Logon window, the change is not activated until the next time the VRPG GUI Designer (in OS/2) or the run-time application is invoked.

Actions are missing from a VRPG project pop-up menu

Restore them with the following procedure (which re-establishes the inheritance link to the VRPG Master Project):

1. Open the settings notebook for the VRPG Client project.
2. Select the **Inheritance** tab.
3. Press the **Add** push button. The **Select a project to inherit from** window appears.
4. From the **Directory** list box, select the ADTS CS for OS/400 folder. If it has not been moved after installation, this folder is under the **DESKTOP** folder.
5. Again, from the **Directory** list box, under the ADTS CS for OS/400 folder, select the VRPG Client folder.
6. The VRPG Client Master Project should appear in the **File** list box.
7. Select the Master Project file, and press the **Inherit** push button.
8. Close the settings notebook for the project.

Some Actions on the VRPG Master Project do not work

This is not an error condition.

VRPG has certain utilities that work on a project's files and directories. Because the VRPG Master Project does not contain any files, these utilities cannot be used on it. Examples of these utilities are: Duplicate, Delete, Rename, Backup, Package, Edit, Run, Debug.

Actions fail after VRPG Create Empty

If you use the **Create Empty** action to create a VRPG Client project to which source files are copied, ensure that the **Source file** name you specify is exactly the same as the source files to be copied. If the name is different, other actions such as Rename or Duplicate will fail.

WorkFrame no longer functions

Several other products such as **Visual Age C++** use WorkFrame to manage their project files. Be aware that part of the process of deleting these projects is to **deregister** WorkFrame objects from the system. Therefore, even though the WorkFrame programs shipped with VRPG Client are not deleted from the installed directory, WorkFrame no longer functions. To correct the situation, you must *re-register* WorkFrame.

To do this, run the command file **EVFICPAM.CMD**.

When it has completed, re-boot the system. This initiates Phase 2 of the VRPG Client installation which re-registers WorkFrame.

Debugger freezes

Applications that contain embedded SQL freeze in Asynchronous mode under the debugger. It is recommended that users running the debugger with embedded SQL always run in Synchronous mode. Answer no if prompted whether the mode should be switched to Asynchronous.

Also if you have MPPM (OS/2 Multi Media feature) installed, the debugger does not work in asynchronous mode.

Insufficient Stack Size error

If you encounter an insufficient stack size system error, you can increase the stack size of your application by increasing the STACKSIZE keyword value in the build options. To change the value, invoke the GUI Designer for the application. On the menu bar, use option "Project", select "Build Options", and in the BUILD Options notebook under the Application Tab, you find entry fields for Stack size and Heap size.

The AS/400 Service Knowledge Base

This can be a useful place to look.

These URLs were correct at the time of writing, but as always may change or disappear at any time.

- STRCODE Problems using OS/2 Optimized Client:

Look at <http://as400service.rochester.ibm.com/sline002/248a.htm>

- Workframe cannot find Action Profiles for a project:

Try <http://as400service.rochester.ibm.com/sline002/251a.htm>

- How to un-install a FixPak:

Check out <http://as400service.rochester.ibm.com/sline002/296e.htm>

RPG Compile and Run-Time Errors

If you get an RPG compile time message error (prefix RNF) or an RPG run-time error message (prefix RNQ), you can look up more details on these messages in the online manual, *RPG Messages*. This online book is located in the VRPG product folder together with the other VRPG online books.

Table of Run-Time Error Codes

When a run-time error occurs, a message box is displayed with a return code value. The following table summarizes the return codes and their meanings.

Code	Description	Cause/Recovery
1	Already initialized	Internal error. Attempt was made to initialize the VRPG environment when it was already initialized.
2	Attribute not valid	Internal error. An invalid attribute was passed.
4	Insufficient resources	Internal error
5	Invalid attribute value	Internal error
7	Invalid event argument	Internal error
8	Invalid handle argument	Internal error
9	Invalid object argument	Internal error
10	Invalid component argument	Internal error
11	Invalid parent argument	Internal error
12	Invalid attribute access argument	Internal error. An invalid Read/Write access was made for an attribute.
14	Invalid type argument	Internal error. An invalid part type was passed.
15	Invalid Valid argument	Internal error. An argument which should contain a true or false value, contains some other value.
16	Invalid value argument	Internal error. The pointer to the attribute value is invalid or not set.
17	Invalid message argument	Internal. The pointer to the message value is invalid or not set.
18	Invalid library argument	This error usually occurs when calling a DLL, and can occur if the DLL could not be found.
19	Invalid function argument	This error occurs when calling a DLL. The function name was not found in the DLL.
20	Not initialized	
21	Part already opened	This error usually occurs when an attempt is made to open a window (SHOWWIN), which is already opened.
22	Part not found	An attempt was made to reference a part that does not exist. This usually occurs when using the fixed form GETATR/SETATR operation code to access a part that is not on the current window.
23	Component already opened	An attempt was made to START a component that has already been opened.
24	Component not found	An attempt was made to START a component, and the component could not be found.
26	Invalid number of arguments	
27	Invalid operation sequence	
31	Language DLL not found	The DLL file for this program was not found.
32	Language DLL entry point not found	Internal error. The action subroutine for an event was not found.

33	Language DLL initialization failed	An internal error occurred when initializing the VRPG DLL.
34	Language DLL version mismatch	There is a level mismatch between the DLL file, and the ODX file. Rebuild the project.
35	Invalid ODX	The ODX file is missing or corrupted. Rebuild the project.
39	Invalid attribute value	Attempt to set an invalid value for an attribute. For example, the attribute value should be numeric, but a string was passed.
40	Event not found	Internal error
41	Missing part registration file	Internal. The file containing valid VRPG part types is corrupted or missing. Reinstall VRPG.
42	Invalid part type	Internal error. An invalid part type has been passed.
43	Error writing to RPG log file	Under certain conditions, RPG attempts to write logging information to a log file. An error occurred writing to this file. This usually occurs if the disk is full.
44	Missing ODX file	The .ODX file was not found. This file should be in the same directory as the .DLL and .EXE files. Rebuild the application.
45	Terminate component request rejected	Internal error. The terminate component request could not be completed.
98	General failure	Internal error
99	Internal error	A unknown VRPG internal error has occurred.

Chapter 11. ADTS CS Notes and Tips

This chapter contains general tips and techniques. The following chapter contains the notes and tips for the GUI Designer.

This list provides a quick cross reference to topics in this chapter:

- “Keeping in Touch”
- “How to Obtain Corrective Service Diskettes (CSDs)” on page 68
- “VRPG Client and License Manager” on page 68
- “Passwords” on page 68
- “Running Two Versions of Application” on page 69
- “Bitmaps and Icons” on page 69
- “Debugging Windows Applications” on page 69
- “Using Disk Caching” on page 69
- “Blocking and Performance” on page 69
- “Setting the VRPG Audio Alarm Option” on page 70
- “Finding the Local LU Details Using CM/2” on page 70
- “Test Directories” on page 71
- “VRPG: Using Local Files” on page 71
- “SQL Bind Files for DB2 Databases” on page 72
- “Back Up and Restore” on page 73
- “WorkFrame” on page 74
- “How to Determine the Service Level of Components” on page 75
- “Removing Win32S” on page 76
- “Specifying the AS/400 USERID and Password” on page 76
- “Communications Troubleshooting” on page 76
- “CODE Tips” on page 80
- “Using ADM” on page 82
- “DBCS Considerations” on page 83

Keeping in Touch

Contact us in one of the following ways:

- Visit the new VisualAge for RPG home page at www.software.ibm.com/ad/varpg/.
- See also the more marketing related Web pages at <http://www.software.ibm.com/ap/as400/adts/adts.cs>.
- On CompuServe, use **GO SOFSOL** to go to the **IBM Software Solutions** area, which lists IBM Software Solutions products. Select VRPG Client from this list to get the most current information.
- Subscribe to the CODE400-I Internet mailing list, which was set up and is maintained by a CODE/400 customer for CODE/400 and VRPG discussions, tip sharing, questions, and so on. While IBM does not own or maintain this mailing list, we do monitor it. To join, send an Internet mail message to **code400-owner@borzak.com** with the subject SUBSCRIBE.
- Send in your Reader Comment form, which you can find in your manuals and online help (select **Help->General help**, then select **Options->Contents**, and finally double-click on **Reader Comment Form**).

How to Obtain Corrective Service Diskettes (CSDs)

PC software other than ADTS CS needs to have CSDs applied just as the AS/400 code and ADTS CS code has PTFs applied. CSDs and the accumulated version of CSDs (a Fixpak) can be obtained from the Internet. Here are some Web Page addresses that allow downloading of CSDs for various IBM products.

OS/2 Fix Packs

Warp V3 should be at FixPak 26 level at least to get the correct WIN32s level. See also "Win32S Run-Time Support: Post-Install Checks and Note" on page 40.

CM/2 Fix Packs

You can download Fixpaks for CM/2 from the following URL:

<http://ps.software.ibm.com/pbin-usa-ps/getobj.pl?/pdocs-usa/allcm.htm>

DB2/2 Fix Packs

You can download Fixpaks for DB/2 from the following URL:

<http://www.software.ibm.com/data/db2/db2tech/version2.html>

VRPG Client and License Manager

VRPG Client now implements license manager checking when the STRCODE command is run on the AS/400 host. If STRCODE does not proceed because of a license manager error message, refer to the *ADTS CS for OS/400 Installation Guide*.

The license manager issues warning messages when STRCODE is invoked if no license information is found or no license is found. It does not prevent STRCODE from running. If STRCODE does not run, please press F10 for other error messages.

Note: Do not get confused when the STRCODE command fails and you see the license manager messages; probably these messages are not pointing to the problem. Ignore them and look at the messages following them to pinpoint the cause of the failure.

Passwords

You cannot use the left and right bracket characters ((or)) in a password, even though the AS/400 system accepts them.

If you are using other language code pages, these are the characters to avoid:

Code page	Code points
Japanese	80, A0, FD, FE, FF
Korean	80, DF, FD, FE, FF
S-Chinese	80, FD, FE, FF
T-Chinese	80, FD, FE

Running Two Versions of Application

You cannot run two different versions of the same application at the same time. An error message stating "Incompatible version" is produced when the second application is started.

Bitmaps and Icons

The best place to keep bitmaps and icons is in the run-time directory of your project. Just use the unqualified bitmap or icon name in the settings notebook.

Putting these files in the run-time directories has the advantage that the packaging tool includes them in the application package it creates so you do not have to worry about building your own files to ship with your application.

Windows bitmaps are different from OS/2 .BMP files and you need a separate set of .BMP files if you support both run-time environments.

Debugging Windows Applications

Remote debugging of Windows applications is a complex task in this release. It requires two machines, one running OS/2, the other Windows with TCP/IP for DOS.

Please refer to the instructions in the V3R6 README file.

Using Disk Caching

The performance of VRPG applications can be significantly improved by using a disk caching program such as **SmartDrive**. A disk caching program uses a portion memory to simulate a disk drive.

Blocking and Performance

Normally blocking is not supported when using the CHAIN operation code and SETLL or SETGT. Not blocking causes decreased performance when dealing with lists of data to be filled from the AS/400 database in a Client/Server environment.

For this reason, VRPG introduced the BLOCK keyword that allows blocking even if CHAIN and SETLL or SETGT are used for a file. Using BLOCK improves the data transfer performance in your VRPG applications.

Note: The database still does not allow blocking when using READE, READPE, and READP operation codes for a file. So you have to restrict yourself to the CHAIN operation code to get the benefits of blocking in VRPG applications.

If you use Blocking and you need the I/O feedback area updated with current information, you have to use the POST operation code to force an update of the I/O feedback area.

See also *VRPG Client for OS/2 Language Reference*, SC09-1847.

Setting the VRPG Audio Alarm Option

VRPG can be set up so that an audible alarm sounds under the following circumstances:

- When **Backup** has completed saving each file.
- When **Restore** has completed restoring each file.
- When **Import** has imported a display file format.
- When the client calls the server to retrieve data for the HAPI server tree.

Opening a branch for the first time forces the client to call the server for data. Subsequent closing and opening of the branches displays the same data again locally.

If you want to have the audio alarm sound, create a file called SOUNDON in the VRPG install directory (usually CODE).

To create the file, you can COPY any other small file. Alternatively, type

COPY CON: d:\SOUNDON

(Substitute your VRPG drive and directory for d:) from the OS/2 prompt. After this, press the spacebar, Ctrl-Z, and Enter.

To disable the alarm feature, delete or rename the SOUNDON file.

Finding the Local LU Details Using CM/2

Note:

Characters marked in reverse image such as **E** are cross references to Table 1 on page 4.

1. In the Communications Manager Folder, find and double-click on Sub-System Management.
2. Select "SNA subsystem" from the list and double-click on it.
3. Select "Display active configuration" from the list by double-clicking on it.
4. Use the PMDSPLAY command from any OS/2 command window.
5. Select [DisplayGeneral SNA] from the menu.
6. Select "Logical Unit 6.2" from the list.
7. The display shows you:

- Details of the LU representing this workstation:

1 LU name	JCPC E
LU alias	TEAR0HA G
Fully-qualified LU name	SYNCR0.JCPC
Default LU	Yes
LU local address	Independent
Configured sessions limit	65535
Transaction programs limit	No limit
LU type	6.2
Number of partner LUs (PLUs)	1

- Details of the Partner LU (your AS/400 system):

1.1 Partner LU alias	5250PLU H
Partner LU uninterpreted name	X'0000000000000000'
Partner LU name	SYNCR0.SYNAS2 A . B
Partner LU session limit	65535
Parallel sessions	Supported
Session security	Not configured
Conversation security	Configured, Active
Already verified security	Not configured
Implicit partner	No
Number of modes for this PLU	2

- Details of the modes configured:

These are normally SNASVCMG followed by QPCSUPP.

8. Now, if you want some *real* detail, trying selecting [Display] [General SNA] [Links] (or [LU6.2 sessions]) instead!

Test Directories

When a VRPG component has been successfully compiled, all of the required run-time files are placed in the RT_OS2 or RT_WIN subdirectories.

These files are *also* copied to the test directories **ICODE\TEST** and **ICODE\WINTEST** (assuming VRPG was installed in directory CODE). The TEST directory is placed in the LIBPATH statement and the WINTEST directory is inserted in the PATH statement of the AUTOEXEC.BAT file when VRPG Client is installed so any component DLL invoked by the START operation code is found when running a VRPG program from the GUI builder.

Be aware that, if you choose to *delete* VRPG Client, the procedure removes these directories. If you require components to be in these directories, you have to rebuild them, or copy all of the project's run-time files to them.

See also "Packaging a Component" on page 25 on some aspects of this topic.

If you decide to copy instead of rebuilding your application, make sure you run the OS/2 command ATTRIB -h *.* in your RT_directory first.

Furthermore, if you *move* or copy a project to another machine, you either have to rebuild the project or copy the TEMP directory entries for it onto the new machine.

See also the discussion on "Packaging a Component" on page 25 in this manual.

VRPG: Using Local Files

In the VRPG F-specification, if a disk file is not specifically designated as REMOTE, it is implicitly local (that is, a PC file, if the EXTFILE keyword is used in the F spec).

1. In the VRPG File specifications, set up a PC file as a program-described disk file.

There are two ways of handling a local file:

- You can work with a fixed record length file. Do not use the RCDLEN keyword.

- You can work with a variable record length file. You have to use the RCDLEN keyword and specify a variable that contains the actual record length for each record you are working with.

The record-length is up to you. But be aware that the Carriage Return/Line Feed pair is inserted every *record-length* characters. This may cause you some extra thought if you are trying to process stream (text-style) information rather than fixed-length record-oriented data.

2. Create I-spec lines for the file name and its record structure.
3. Open, Close, and sequential I/O operations are just as for AS/400 files.

If the PC file exists in the path of your application at run time, you do not have to do extra work. If the file is *not* found at run time, it is created (in the run-time directory of the VRPG application). If you use the EXTFILE keyword, you can move the path information and real file name into the variable specified with the keyword and VRPG uses this information to access the correct file at run time.

If you do not use the EXTFILE keyword, you need to tell VRPG through the [AS/400 information] dialog the specifics of the file you want to access:

1. First make an entry for the file in (believe it or not) the AS/400 Information notebook of the VRPG Client dialog:
 - Use **[Project] [Define AS/400 Information]**.
 - Select the **Files** Tab of the settings notebook.
 - Press the **Add** push button.
 - a. In **File Alias**, enter a name to be used in the VRPG F-spec for the file.
 - b. In **Remote File**, enter the actual name of the file as it is found on the PC.

If you do not enter the path with the file name, the file is assumed to be in the run-time directory of the application. In the development environment, this is the RT_OS2 directory; in an installed run-time environment, it is the directory containing the executables.

If the file is not found, it is created at OPEN time, whether it is an input or an output type.

- c. From **Server Alias**, select ***CLIENT**.
- d. Press **OK** (or **APPLY**, if you want to define further files now).
- e. Dismiss the **AS/400 Information** notebook.

If you want your local file to have a fixed record length instead of a variable record length, do not use the RCDLEN keyword in the F spec of this file. A fixed record length has the advantage of providing faster performance when using access by relative record number. With a fixed record length, the record location can be calculated instead of being forced to read sequentially through the file until the right record can be found for variable record length.

SQL Bind Files for DB2 Databases

If you build a project containing embedded SQL statements using the default name for the bind file, build it again using a different bind file name. There are two bind files in the run-time directory.

If you are packaging the project, both bind files are packaged because previously created bind files of different names are not deleted from the run-time directory.

You must remove the unnecessary bind files from the project's run-time directory.

Back Up and Restore

Information regarding Backup and Restore of VRPG projects to an AS/400 system is collected in the following section.

Backup Notes

- The Backup function creates files on the AS/400 system based on the project file name you specified when the project was created. Since OS/2 has a different set of rules for creating file names, you should ensure that the project filename you specify follows AS/400 naming rules.

For example, a file name of 12345678 is valid for OS/2, but not for the AS/400 system.

- When VRPG projects are backed up to the AS/400 system, file attributes associated with user files are not preserved. When the project is restored, you have to reset the attributes.
- Backup does not back up the build options for the project. Therefore, if the project is restored to a different location on the same machine, or to a different machine, the build options have to be specified again in the GUI Designer.
- VRPG backs up a project to a library that you specify on the AS/400 system.

The project consists of two files created in the library:

1. A source physical file, which holds all of the project source such as the VPG file.
2. A physical file, which contains all binary files, such as the project's .EXE file.

These files are created using the eight-character file name you specified when the project was created.

This name is appended with "SF" to create the source physical file, and with "DF" to create the physical file.

For example, if a project was created with the name SAMPLE, a source physical file named "SAMPLESF" and a physical file named "SAMPLEDF" are created in the library.

- During a backup, these files are created if they are not found.

If the files *do* exist, a message is displayed indicating that they are overwritten by the backup.

If only *one* of the files exists (if, for example, one of the pair has been deleted, or there is a user file with the same name as the backup file), the backup process ends.

To correct this, you must either re-create the missing file, or delete the existing one.

Restore Notes

The Restore function displays only the physical files (PF-DTA) from the library selected.

To see only VRPG backup files, subset the list such that only files ending with "DF" are displayed:

Use an entry similar to this:

5250PLU TESTLIB/*.DF

in the entry field of the Restore Project window to subset the selection list.

When a file is selected in Restore, VRPG reads the header information created by the Backup function from the file to ensure it is a VRPG project backup file, and not a user file.

- When a project on the AS/400 system is selected to be restored, a dialog appears that indicates to which directory the project file and the project's source files are to be restored.

WorkFrame

The next sections contain Workframe specific hints and tips regarding VRPG projects.

Projects and Composite Projects

- Do not change the first source directory or the target directory in the settings pages of a VRPG base project after it has been created. VRPG assumes that the first directory is where the necessary files are in order for it to run (that is, the .ODF file).
- Do not change the target program in the settings page of a VRPG base project. It is this name that is used to find the necessary files in the first source directory that are needed to open the project.
- Do not have more than 25 items in a Composite project unless you have the fix for APAR PJ15618 installed on your system. Having more than 25 items without this fix applied may result in unpredictable results.

If you do not have this fix and you begin running into trouble when opening the composite project, try going to the file system and moving the base project files out of the directory that represents that composite project.

- Do not change a base project when it is currently open. Performing such operations as rename and duplicate may result in unpredictable results if they are invoked while the project is open.

Renaming Projects

The **Rename** function of VRPG renames all of the files created by VRPG to the new name you specify. Be aware that the file with the **VPF** extension contains records that refer to the original name. You must manually change these records to refer to the renamed file name.

The same occurs if you use the **Duplicate** function.

Tools

It is highly recommended that if you want to copy, rename, or delete VRPG projects, use the tools supplied by VRPG and not the ones supplied by the WorkPlace Shell or the file system. For example, if you rename the ODF file using the system rename, VRPG cannot work with that project again until it is renamed back.

There is a tool found in the VRPG Client/2 folder called **WP Class List**. This tool shows a list of the WorkPlace Shell object classes currently registered in the OS/2 WorkPlace Shell. It has been added to our tools so that if you run into any problems registering any VRPG objects, you, with the help of support personnel, can register them using this tool.

This tool can also be used to deregister object classes.

Moving Project Files

If it becomes necessary to move VRPG project files (for example, to free up disk space), you must perform the following steps on the WorkFrame folder for the project so the VRPG client can successfully locate and open your project:

1. Open the settings notebook for the project.
2. Select the **Location** tab.
3. Change the three directory entries listed so they now reference the directory where the files were moved.
4. Ensure that the first directory listed is the one that contains the project source files.
5. Ensure that the target program specified has the same name as the source file name minus the extension.
6. Ensure that the **Working directory** value on the **Location** page is `..\RT_OS2` or `..\RT_WIN`.
7. Close the settings notebook.

Note 1. Be aware that some VRPG project files have the **hidden** attribute set. When copying project files, be sure to specify the `/h` command switch to include hidden files.

Note 2. Do not rename any of the project files independently. To rename a project, use the **Rename** action from the pop-up menu.

How to Determine the Service Level of Components

- Open the "ADTS CS for OS/400" Folder.
- Select "Installation and Maintenance".
- Click on "ADTS C/S for OS/400".
- Select [Details] [Product Status] from the menu.
 - The version of the ADTS CS Product is shown at the top of the display.

The individual installed components are listed below.

- Click on the component of interest.
- Click on the "Service Level" push button.

Removing Win32S

This was raised on the VRPG CFORUM in response to a problem where WIN32S hung. The suggested work-around was to remove Win32S and manually re-install it. Failing that, re-install Windows and the run time.

1. Delete W32SYS.DLL and WIN32S16.DLL from the windows/system directory.
2. Delete all of the files in the windows\system\win32s directory.
3. Remove the following entry from the ENH386 section of SYSTEM.INI:
device=c:\windows\system\win32s\w32s.386
4. Re-boot.

Specifying the AS/400 USERID and Password

In development, these values can be entered into the **Define Server Logon** window in the GUI Designer from the Project menu item.

On run-time platforms, click on the Define Server Logon icon to define the users for that client.

You can also call FVDEPW from a command line.

In any case, the information is stored in a hidden file with the Password encrypted.

Communications Troubleshooting

The following list contains some of the things to check to help resolve VRPG communication failures with the host (AS/400 system).

On the host side:

1. Check if QIWS library exists with program modules QEVxxxxx in it.
2. Check if PTF SF20554 (5763SS1) is applied (communication fixes).
3. Check if PTF SF22234 (5763VR1) is applied.
4. Check if subsystem QBATCH is active.
5. Check if subsystem QCMN is active.
6. Use WRKCFGSTS *LIN to determine if the communication line, controller, and device are active.
7. Use WRKACTJOB, DSPSBSD, and WRKJOBQ to determine if the remote communication request is being processed or queued up.
8. Check the QSYSOPR message queue for remote logon failure messages for your PC.
9. Check the spooled file (WRKSPLF, WRKSBMJOB) for job log relating to the communication link up attempts.
10. Try to execute STRCODE from the 5250 emulator to the PC. If this fails, check if the required communication file exists (communication file name should be EVFCICFF, see the installation manual for the setup procedure).
11. Check to see if the required communication file exists in the library in the library list specified in your job description.

Note: The library containing the required communication file should not be in the batch job description in QGPL. If so, some other user may accidentally use your communication file for their remote communication

link. Use the DSPUSRPRF command and WRKJOB command to determine the batch job description, current library, and library list being used.

On the Workstation side:

1. Check if the Communications Manager has been started.
2. Check the error log FVDCERRS.LOG for hints on communication failures. This lives in the project's run-time directory for installed applications, or in the CODE directory in development.
3. Check if an alias name is used for establishing communication.
4. Check FVDCSEC.TXT to make sure entries (password, alias name...) are specified correctly.
5. Check to make sure the remote user ID, password, and alias name are eight characters or less.
6. Check that the Partner LU (PLU) alias used in the Communications Manager setup matches the name used in establishing the communication. Use upper case in the Communications Manager entry and logon request to make sure the upper/lower case matches.
7. Check the .RST file contents to make sure the default and specific communication entries are correct.
8. Try to activate the 5250 emulator to the host. If this fails, check the address defined for the host in Communications Manager. The AS/400 host address can be found by using the WRKCFGSTS *LIN command on the host to display the line description used for the workstation communication.

Running a Communication Trace on the AS/400 System

If you suspect the problem is on the host side, running a Communications Trace on the AS/400 system shows the data that is leaving and being sent to the system. The following steps create a communication trace on the AS/400 system.

1. Log on to the AS/400 system.
2. Determine the name of the line to be traced. If not known, use the **WRKCFGSTS** command and specify *LIN as the type to see defined communication lines.
3. Enter **STRSST** on a command line.
4. Select option 1 **Start a service tool**.
5. Select option 3 **Work with communication trace**.
6. Press the **F6** command key to start a trace.
7. On the **Start Trace** display:
 - For **Configuration object**, specify the name of the communication line determined in the first step.
 - For **Type**, specify 1 (Line).
 - Type any description for the **Trace description**.
 - Set the buffer size to **6**.
 - Set **Stop on buffer full** to **Y**.
 - Set the **Data direction** to **3** (both).
 - Press Enter to start the trace.
8. Run the communication scenario until the communication failure occurred.
9. On the **Work with Communications Traces** display, select option 2 to stop the trace and select option 6 to **Format and print** the trace.
10. When the dump is finished, go back to the command entry.
11. Enter WRKJOB.
12. Take option 4 to work with spooled files.
13. The trace dump spooled file is listed here.

Running a Communications Trace on OS/2

If checking the preceding suggested areas did not resolve the problem, collect a communication trace from the host and the PC for investigation.

1. Open the **Communications Manager** folder and double-click on the **Problem Determination Aids - Trace** icon.
2. Choose **Options** from the menu bar and choose sub-item **Trace buffer defaults...** and set the defaults to their maximum values. This avoids buffer wraparound.
3. Select **File** from the menu bar and choose **Clear trace buffer**.
4. Turn your attention to the three list boxes in the middle of the window. In the APIs list box, select "APPC" and "CPIC" (if available). In the "DLCs" list box, select "IBMTRNET" (assuming you are using token-ring). You do not need to select any entries in the "Events" list box.
5. Type any name you want in the "Trace filename" entry field (the convention is to use the extension .TRC).
6. Click on the "Start" pushbutton.

Now that you have the trace up and running, retry the failing scenario. When the failure occurs, go to the **Trace Services** window and complete the trace collection:

- a. Click on the "Stop all" button.
- b. Click on the "Save" button.

There, that's it. Now you can upload the trace in binary and send it to development for analysis.

Communication Trace on Windows 3.1

This section documents the steps for setting up the communication trace tool for the Windows 3.1 environment.

Note: Client Access for Windows must be installed with the Windows 3.1 operating system.

Installing NSTRACE

Use the following steps to install the Network Services Communication Trace Function (NSTRACE):

1. Start up the PC Client Access connection to an AS/400 host with CA/400 installed.
2. Sign on to the host by clicking on the connection icon.
3. After the successful logon, select the Folder icon to connect the I: drive to \\host\QIWSTRTOOL folder. "Host" is the alias name of the host you are connecting to.
4. Start a Windows command prompt.
5. Go to the I: drive.
6. Enter the IWSTRTOOL command.
7. Select NSTRACE from the list.
8. Read the instruction on how to set up the NSTRACE function.
9. Go to the bottom of the file and enter the directory path where you want to have the NSTRACE tool installed (for example, C:\NSTRACE).
10. Press Enter and the tool is downloaded to the specific directory.

Setting Up the NSTRACE Tool

Perform the following steps to set up the trace tool once it has been installed:

1. Edit the NSTRACE.CFG file in the directory where the tool was installed. Update the existing entry C:\CAWIN\NSTWTRC.CFG to indicate the current path where the tool was installed (for example, C:\NSTRACE\NSTWTRC.CFG).

The entry C:\CAWIN\MISC\ANYTRACE.CFG is valid only if AnyNet is installed and used in CAWIN. In that case, make sure the sub-directory MISC exists under the CAWIN directory where CA for Windows is installed.

2. Go back to the Windows session and create an icon in the CAWIN folder for the NSTRACE function. Change the properties of the new icon to point to the NSTRACE.EXE file in the directory where the NSTRACE tool was installed.

Using the NSTRACE Function

Once the NSTRACE function has been installed and set up, it can be run as follows:

1. Click on the NSTRACE function icon.
2. Select the Setup pull-down.
3. Select the Initialize function from the pull-down.
4. Select the Enable Automatic Buffer Dumping option and click on the OK button.
5. Click on the OK button on the "Initialization completed" message.
6. Select the Setup pull-down again.
7. Select the Configure... function.
8. Select the following components and their corresponding trace options:
 - CPIC-C and all its associated trace options
 - EHNAPPC and all its associated trace options
 - LAN and all its associated trace options
 - LU and all its associated trace option, if any
 - NOF and all its associated trace option

When finished, click on the OK button.

Note: These examples are for a LAN setup. If a different setup is used, select the relevant components and trace options (see Client Access documentation for more details).

Starting the Trace

1. Select the operation pull-down in the NSTRACE icon.
2. Click on the Start Trace option.

The communication trace is now started. Run the scenario you want to trace.

Collecting the Trace Information

1. When the scenario to be traced is completed, go back to the NSTRACE icon where you started the trace. From the Operation pull-down, select **Suspend Trace**, and then **Format Trace**.
2. When the formatting is completed, the trace file FMTNS.LOG is in the working directory for the NSTRACE (refer to the information entered in **Properties** when creating the icon for NSTRACE). The default working directory is the directory where the NSTRACE tool was installed.

CODE Tips

This sections contains some hints and tips about commands available in CODE/400.

The CODESRV Command

With CODE/400 and VRPG Client comes an OS/2 command (CODESRV) for accessing the AS/400 system. Using this command in your CODE editor macros or REXX macros can really make a powerful combination. You can retrieve lists of names of AS/400 CODE servers, libraries, objects, files, members, records, fields, and keys as well as ADM/400 projects, groups, and parts.

You can also execute AS/400 commands and copy source members back and forth from the AS/400 system to your workstation.

To find what CODESRV can do, from an OS/2 command line, type:

```
CODESRV /H
```

Use the CODESRV command to:

- Shut down one or more active host CODE server (or servers).
- Get a list of active host CODE servers.
- Send a command to a CODE server.
- Specify CACHE support.
- Get message descriptions.
- Download a member from a host server to your workstation.
- Upload a file from your workstation to a host server.
- Get a list of objects that match the task you specify.

Use the following syntax. Note that the opening round bracket is the parameter list marker and is required if any of the keyword parms are used.

```
CODESRV ADMLIST server adm-list (ADMDESCR ADMPCODE ADMLGLDATE ADMLGLTIME  
CACHE REFRESH
```

```
CODESRV EXEC server OS/400-command (SWITCH LOG
```

```
CODESRV GET server member filename (NOTAG
```

```
CODESRV HOST server
```

*{Entry for server may be '*FIRST'}*

```
CODESRV LIST server-list (ATTR CACHE DATE DESC LVLCHK LENGTH TYPE  
REFRESH
```

{REFRESH must be preceded by CACHE}

```
CODESRV MSG MsgId MsgFile (SECOND CACHE CODE REFRESH
```

*{REFRESH must be preceded by CACHE}
{Use QCODE/EVFCMSGF}*

```
CODESRV PUT server filename (NOTAG REPLACE  
CODESRV PUT server filename member (NOTAG REPLACE
```

```
CODESRV SERVER
```

```
CODESRV SHUTDOWN
```

{Or CODESRV /S}

CODESRV SHUTDOWN server (LOG

Some Examples:

- To get a list of all the active servers on AS/400 hosts, type:
CODESRV SERVER
- To print the LONDON source member using SEU's option number 6, type:
CODESRV EXEC OS400 STRSEU OPTION(6) SRCFILE(TRAVEL/ENGLAND)
SRCMBR(LONDON)
- To browse the LONDON source member using SEU's option number 5, and switch to the emulator window that runs the command, type:
CODESRV EXEC OS400 STRSEU OPTION(5) SRCFILE(TRAVEL/ENGLAND)
SRCMBR(LONDON) (SWITCH
- To get a list of all the members contained in a source physical file named ENGLAND in the TRAVEL library on the OS400 server, type:
CODESRV LIST OS400 'TRAVEL/ENGLAND(*)'
- To put this list into an OS/2 file, type:
CODESRV LIST OS400 'TRAVEL/ENGLAND(*)'
LIST.TXT

The list is added to an OS/2 file named LIST.TXT. (The greater than symbol (>) directs output that normally appears on an OS/2 window to a file.)
- To get a copy of a member named LONDON contained in a source physical file named ENGLAND in the TRAVEL library on the OS400 server and copy it to a file on your workstation called LONDON.MBR in your TRAVEL directory on your C drive, type:
CODESRV GET OS400 'TRAVEL/ENGLAND(LONDON)'
C:\TRAVEL\LONDON.MBR
- To put the LONDON.MBR file back on your server and replace the existing member, type:
CODESRV PUT OS400 C:\TRAVEL\LONDON.MBR
'TRAVEL/ENGLAND(LONDON)'
- To shut down all of the active servers on AS/400 hosts, type:
CODESRV SHUTDOWN
- To shut down just one server named OS/400, type:
CODESRV /S OS/400

The EVFCCMD Command

To view the CODE command log, type the following command on a command line:

EVFCCMD

The CODEEVNT Command

To view a CODE events log, use the following syntax from the command line:

CODEEVNT filename

Examples

- To open the event file in the library (TRAVEL) on the AS/400 host that is accessed by the default server, type:

```
CODEEVNT "<>TRAVEL/EVFEVENT(SRCTXT)"
```

The event file is always called EVFEVENT and is created in the object library by the compiler.

The event file member (in this case, SRCTXT) has the same name as the source member you were compiling.

- To open the event file in a ADM/400 project with the short name TRAVEL in an ADM/400 group with the short name GROUP on the AS/400 host accessed by the default server, type:

```
CODEEVNT "<>TRAVEL.GROUP/EVFEVENT(SRCTXT)"
```

The event file is always called EVFEVENT and is created in the object library by the compiler.

The event file part (in this case, SRCTXT) has the same name as the source part you were compiling.

For further information, refer to the online reference "CODE Command-Line Invocation".

The CODEBRWS Command

To browse ASCII files using the CODE Editor (LPEX), use the following base syntax from the command line:

```
CODEBRWS filename
```

There are several optional parameters. Please see the command line reference.

CODEBRWS calls the following OS/2 executable program:

```
EVFXLXPM.EXE /CODE /V
```

Other Neat Things in CODE

Read the following online books provided with the CODE/400 feature:

- CODE Read-Me-First (EVFREAD.inf)
- CODE Tips & Techniques (EVFFAQ.inf)
- CODE and VRPG EXTRAS function (EVFEXTRA.inf)
- CODE Editor "How Do I...?" (EVFELHDI.inf)
- CODE Command-Line Invocation (EVFGS400.inf)

Using ADM

Some information on ADM and its support for VRPG and Code/400 is provided in the following sections.

Functions Supported

VRPG supports the Application Dictionary Manager (ADM) functions of CHECKIN, CHECKOUT, and EXTRACT. This support is only available for Version 3 Release 2 of OS/400 and later on AS/400 CISC systems, or Version 3 Release 6 and later on AS/400 RISC systems.

The ADM functions apply to individual projects only. For example, if ProjectA contains ProjectB and ProjectA is checked in, ProjectB is not automatically checked in. Therefore, if multiple users are working with the ADM functions to manage a VRPG project, they must ensure that the project structure is the same on all workstations.

Error Feedback for CODE/400 with ADM Builds

To get CODE/400 error feedback after doing an Application Development Manager (ADM) build, you must first create an Application Development Manager part of type BLDOPT with the same name as the part being built.

The part should be in the same group, or higher, as the part being built.

If it is an Original Program Model (OPM) compile or preprocessor command, in the BLDOPT part, uncomment the appropriate compile command and add OPTION(*SRCDBG) or OPTION(*LSTDBG) to the command.

For ILE compile and preprocessor commands, add OPTION(*EVENTF) to get error feedback, and DBGVIEW(*SOURCE) or DBGVIEW(*LIST) to enable the CODE debug tool.

Note:

1. You may want to have a different BLDOPT part in the production group, one which does not specify these options.
2. Create a BLDOPT part named QDFT to have error list feedback available for all parts being built. This avoids having to create a BLDOPT part for each part.

DBCS Considerations

Use the following considerations if you plan to run VRPG Client/2 on a DBCS system.

- VRPG does not allow shift-in, shift-out characters in literals. Therefore, if you use LPEX to open an AS/400 member for edit to copy source into your VRPG program, you must remove the shift-in, shift-out characters from all literals. If they are not removed, compile errors occur.
- If you are installing VRPG Client/2 on OS/2J Version 2.1, you must have CSD Service Pack BJC0006 or later installed.
- If you are installing VRPG Client/2 on OS/2J Version 2.11, you must have CSD Service Pack MJC0002 or later installed.

DBCS Code Page Support

If you are using the VRPG Client/2 product and CODE/400, consider:

OS/400	OS/2
J 930 (ext. Katakana UDC 4370)	932/942
J 939 (ext. English UDC 4370)	932/942
J 5026 (ext. Katakana UDC 1880)	932/942
J 5035 (ext. English UDC 1880)	932/942
K 933 (ext.)	944/949
S 935 (ext.)	1381
T 937 (ext.)	938/948/950

There are differences between the characters in EBCDIC code pages and PC (ASCII) code pages. The following table summarizes some of these differences.

CCSIDs	Missing characters
932 (J-PC)	Back slash, cent, logical not, tilde, pound
949 (KS code)	Back slash, cent, logical not, vertical broken line
938 (T-PC)	Cent, logical not, vertical broken line
950 (Big-5)	Cent, logical not, vertical broken line

Note

- If you want to use the characters that are not available for CCSID 949, it is recommended that you use CCSID 944.
- The CODE/400 server can only be started with the preceding list of code pages or CCSID support.
- If you want to develop client/server applications and avoid the discrepancies between OS/2 and OS/400 code pages, use SAA code pages.

Chapter 12. GUI Designer Notes

This chapter contains a loose collection of notes, tips, and errata from the manuals, READMEs, and forums.

Project Operations

Be careful and use only the VRPG supplied commands when dealing with projects do not operating system commands for projects.

Project Rename

Do not rename a project while it is being edited by the GUI designer. If you do, unpredictable results may occur.

Project New, Project Save

Save any changes to subroutines before you use Project New, Project Open, or Project Save. Otherwise, unexpected results may occur.

Entry Field Parts Settings Notebook Behavior

The following sections contain some tips for using the Settings Notebook for Entry Field Parts.

Specifying Range of Values for Entry Field

If you have an entry field defined with, for example, a field length of five and also want to specify a range for validation, you might get an error indicating that the values you specified are not valid. It might not be obvious to you why this error gets generated. Probably the length of the values you specified in the Range notebook page is longer than the field length you specified. For example, for a character field with a length of five, you specify 111111.

To recover, make sure that you correctly specify the range values and that they do not exceed the field length of the entry field.

Changing Entry Field or Static Text Part to Numeric

If you have default character values specified for an entry field or static text part and change the definition of this part to numeric, you get an error indicating that the specified text is not valid. Remove the character data from the Text entry field on the General page of the parts setting notebook to avoid this error.

Note: The preceding error may also happen if you have blank characters defined in the Text entry field on the General page of the parts setting notebook and it is not easy to spot what the cause of the error message is in this situation. Make sure the field is really empty and no blank characters are hidden.

AS/400 Information Notebook

In the **Define AS/400 Information** notebook under the **DataAreas** tab in the **Remote Name** column, only the first 10 characters of this field are recognized.

The help indicates that the syntax for this field is **library/dataarea**. However, only the data area name should be entered.

Changing Action Links

In Release 1, the **Action Link** window was accessed from the GUI Designer menu bar. This function is now accessed from the editor. To display the Action Link window from the editor, select **Edit** and then **Action subroutines**.

Message Box After Restore

Do not create the following sequence within one action subroutine:

1. Restore a window from minimized state.
2. Immediately display a message box.

This causes the message box to appear briefly and then disappear. The window appears without focus. The focus for the window cannot be regained and the user interface for the application is frozen.

Message Description Length

In the Define Messages window, you can add a description. The multi-line edit wraps the text as you type. However, if you back up and restore your project, you lose any text typed beyond 132 characters. To avoid this, manually format your comments by periodically pressing Enter to wrap your text line-by-line.

Technical Descriptions

Some of the user text entered in the Technical Description of a part may be truncated during backup/restore.

On the notebook settings page for any part, the user is allowed to type comments into the technical description field. Since this is a multi-line edit, it automatically folds lines. However, the actual text is stored as a single flat file, and usually any line longer than 132 characters is truncated when the project is backed up.

To avoid this when typing text into the technical description field, press the Enter key once in awhile to format each new line. By pressing Enter, a new line character is added to the text, creating separate records that can be safely saved on the AS/400 system during backup.

Windows Run Time: General Differences

Specific differences are documented later under the parts, events, and attributes affected.

Number of Parts

Windows has a restriction on the number of window parts that can exist at one time. If you have an application that creates many parts, you may experience run-time errors.

To minimize the number of parts that are created, consider using the SHOWWIN and CLSWIN operation codes that create and remove windows in memory, if necessary.

No Second-Level Help

Second-level help for messages cannot be displayed in Windows. If a message box is displayed with a message that has second-level help defined, the **Help** push button is not on the message box.

No Drag-and-Drop

Drag-and-drop is not supported as documented in the *Parts Reference*.

Win-OS2 Run Time: General Differences

Specific differences are document later under the parts, events, and attributes affected.

Font Sizes

Due to the differences in font sizes between Windows and WIN-OS/2, the cursor may not always appear in entry fields. Data can still be typed in the entry field. To circumvent this problem in WIN-OS/2, increase the height of the affected entry field, or choose a smaller font.

Events

The following sections contain information about VRPG Events.

Close

The Close event is not documented. This event occurs when the user closes a window by selecting Close from the system menu of the window. You should use this event to determine if your application should be terminated or not.

A Close event is only generated when a window is closed using the windows **System** menu; it is *not* generated by the CLSWIN operation code.

Destroy

In the first release of VRPG when a window was closed, a Destroy event was signalled for that window. If other windows were still active, they were referenced during this destroy event.

This release of VRPG introduces the new **Terminate on close** window attribute, which is set on by default for the first window created for a new VRPG component. When this window is closed signalling that the component is to end, windows are destroyed in the reverse order that they were created. Therefore, during the destroy event for this window, other windows in the application may have already been destroyed so they cannot be referenced at this time.

The lowest children are destroyed first. For instance, for a window that has a canvas that has a notebook that has a notebook page that has a push button, the order of destruction is:

1. Push button
2. Notebook page
3. Notebook
4. Canvas
5. Window

Also note that the destroy event occurs before the part is actually destroyed (removed from memory).

Resize

It is not recommended that message boxes be displayed during a window Resize event. The application may terminate prematurely.

Terminate on Close

See the preceding *Destroy* event.

If you create cascaded application projects, closing one of the cascaded projects closes the entire application, not just the cascaded one. This is because the default state for the **Terminate on Close** attribute for the first window of a project is enabled.

VRPG Client treats cascaded projects as DLLs at run time.

Attributes

The following sections contain information about VRPG part attributes.

Color Attributes in Windows Run Time

These parts do not support the **BackMix**, **ForeMix**, **BackColor**, or **ForeColor** attributes under Windows:

- Combination box
- Group box
- Multi-line edit
- Push button

Font Attributes in Windows Run Time

The **FontBold**, **FontItalic**, **FontUnder**, and **FontStrike** attributes are not supported under Windows.

Height and Width in Windows Run Time

The **Height** and **Width** attributes cannot exceed the maximum height and width determined by the display resolution used.

For example, you cannot create a part 800 pixels wide when working with a 640 pixel width display.

Mouse Focus in Windows Run Time

The **MouseFocus** attribute has no effect during Windows run time. This attribute applies to the check box, graphics, push button, and radio button.

Multiplier Attribute

For the Multiplier attribute, GETATR cannot be used at run time. In the *Parts Reference*, the examples show GETATR operation codes. These examples are not correct.

MultiSelect and ExtSelect

The *Parts Reference* states that **MultiSelect** must be enabled before **ExtSelect** is. This is not true. The **MultiSelect** attribute has no effect on **ExtSelect**. The **ExtSelect** attribute overrides the settings for either **MultiSelect** or single selection.

Extended selection allows the user to select multiple items (two methods are allowed):

1. Select one item and add more items by pressing the CTRL key while selecting them with the mouse.
2. Select by dragging the mouse cursor across the items to be selected.

Note: A single mouse select without pressing the CTRL key de-selects all previously selected items.

This is different from multiple selection where dragging is not allowed and a single click on items selects without de-selecting already selected ones. There is no need to press the CTRL key while selecting.

Outline Font Attribute

It is possible to set this font attribute through the settings notebook or by dropping the attribute from the Font palette. Be aware, however, that the outline attribute cannot be set at run time.

SelectIdx and Groups

For the **SelectIdx** attribute to return the correct radio button index, only radio buttons should be in a group. Adding other part types to the same group may result in erroneous values returned.

Display Size System Attributes

System attributes, **%DspHeight** and **%DspWidth**, are described in the *Client/Server Visual Programming with VRPG Client for OS/2*.

These attributes allow you to retrieve the width and height in pixels of the display on which your application is running. These attributes must be defined on a **D** specification.

One use of these attributes is to reposition windows so they are always in view, regardless of screen resolution. The following code sample ensures that a window is vertically and horizontally centered on the display.

```
DName+++++ETDsFrom+++To/L+++IDc.Keywords+++++
D %DspHeight      S              4P 0
D %DspWidth       S              4P 0

CLO1N01Factor1+++++OpCode(E)+Factor2+++++Result+++++Len++D+HiLoEq
*
* Get windows width and height
C      'WIN1'      getatr  'Width'      winWidth      4 0
C      'WIN1'      getatr  'Height'     winHeight     4 0
C      %DspWidth   sub     WinWidth     widthDiff     4 0
C      %DspHeight  sub     WinHeight     heightDiff     4 0
C      widthDiff   div     2             widthDiff     4 0
C      heightDiff  div     2             heightDiff     4 0
C      'WIN1'      setatr  widthDiff     'Left'
C      'WIN1'      setatr  heightDiff    'Bottom'
```

Parts

The following sections contain additional information about VRPG parts.

Audio

Compression types

The audio part only processes non-compressed wave (.WAV) files. Some wave files are shipped in a compressed format. The audio part cannot process these files.

MMPM/2

For the audio part to work in the OS/2 environment, the multi-media extensions (MMPM) must be installed on the workstation. If this support is not installed, setting the audio part attributes has no effect, and no error is indicated.

Record function

The Record function does not work for the audio part in the Windows environment.

Volume attribute

The Volume attribute has no effect on the audio part under Windows.

Check Box

MouseFocus

The MouseFocus attribute has no effect during Windows run time.

Combination Box

Color

When resetting the color of a combo box or slider part to the system default, the color is not visibly changed until you close and reopen the window.

The following color attributes are *not* supported under Windows:

- BackMix
- ForeMix
- BackColor
- ForeColor

Drag

All types of combo (simple, drop-down, drop-down combination) support dragging from the entry field portion. Only the Drop-down-list type supports dragging from the list box portion (the simple type and drop-down type do not). All variations support dropping onto the entry field portion only (not the list box portion).

Drop Down List type in Windows

In the Windows 3.1 run time, there is no **DropDownList** type for a combination box. If you set it to this type, it defaults to the **DropDown** type. If you want to emulate the **DropDownList** type, set ReadOnly to 1.

ReadOnly

The Parts Reference manual incorrectly indicates that this attribute can be set in the notebook settings of the Combination Box part. It cannot.

SetItem and Sequence

Contrary to what is documented in the *Parts Reference*, the **Sequence** attribute does not affect **SetItem**.

Component Reference (CRF)

Notify event

For a Notify event to be signaled, an action subroutine must exist for the event that is being monitored, even if it contains no code. If an action subroutine is not coded, the Notify event does not occur.

Container

Title

If the title of a container column contains blanks, the container is not displayed correctly. Replace all blanks in the title with the underscore character (_) to correct this.

Click event

The Click event is not signalled for a container part that is in Details view.

MultiSelect and ExtSelect

The *Parts Reference* states that **MultiSelect** must be enabled before **ExtSelect** is. This is not true. The **MultiSelect** attribute has no effect on **ExtSelect**. The **ExtSelect** attribute overrides the settings for either **MultiSelect** or single selection.

DDE Client

Format attribute

The DDE Client part supports a format attribute. This attribute is not documented in the *VRPG Client/2 Parts Reference* manual.

Use this attribute to indicate to the server application the type of data that is being referenced by the **Item** attribute. The format of this attribute is determined by the server application. This attribute is of type string, and should be set before using the **POKE** attribute.

Not available in Windows run time

DDE Client is not supported by the Windows run time. The **DDE Client** Samples and **DDE Client** Hot Link Samples in the *Parts Reference* manual are for OS/2 only.

Entry Field

Alignment attribute in Windows

Under the Windows run time, the alignment attribute of an entry field part cannot be changed once the part is created.

AutoScroll attribute in Windows

The AutoScroll attribute is not supported in the Windows run time.

Color attributes in Windows

If the color is changed from its default value, the new color is ignored at run time and the part is displayed with its default colors.

Edit codes

The edit codes allowed on the settings notebook for the entry field perform the same function as described in the *VRPG Reference* manual.

InsertMode attribute

The InsertMode attribute is not supported for the entry field part in the Windows run time. Changing the **InsertMode** attribute is only allowed in the OS/2 run time.

Press event under Windows

In the Windows run time, if the cursor is in an entry field and the Enter key is pressed, the currently highlighted push button generates a PRESS event even if the entry field has focus. You can determine which push button should be activated by setting the **Highlight** attribute for the push button when the entry field receives the **GOTFOCUS** event.

Graphics

MouseFocus

The MouseFocus attribute has no effect during Windows run time.

Graphic Push Button

HelpEnable under Windows and Win-OS2

The HelpEnable attribute for the push button and graphic push button parts does not work in the WIN-OS/2 or Windows run-time environments. To display the help, give the part focus and press **F1**.

Group Box

Color in Windows

The following color attributes are *not* supported under Windows

- BackMix
- ForeMix
- BackColor
- ForeColor

Image

Clearing the image

To clear the image from an image part, specify a null string as the file name.

List Box

Color attributes in Windows

If the color is changed from its default value, the new color is ignored at run time and the part is displayed with its default colors.

MultiSelect and ExtSelect

The *Parts Reference* states that **MultiSelect** must be enabled before **ExtSelect** is. This is not true. The **MultiSelect** attribute has no effect on **ExtSelect**. The **ExtSelect** attribute overrides the settings for either **MultiSelect** or single selection.

During run time under Windows, the **MultiSelect** and **ExtSelect** attributes cannot be changed for a list box part.

SetItem and Sequence

Contrary to what is documented in the *Parts Reference*, the **Sequence** attribute does not affect **SetItem**.

Scroll Bars

For the List Box and Message Subfile parts, vertical scroll bars behave differently for OS/2 and Windows:

- In Windows, the vertical scroll bar appears only where there are sufficient entries.
- In OS/2, the scroll bar always appears, but is only enabled when there are sufficient entries.

Menu Item

Split style

When a long pull-down menu is created, and the style of a menu item (somewhere in the middle of the pull-down) is set to "split" or "split with separator", the previous menu item's notebook may not be opened with the mouse.

Use the keyboard instead.

Button separator style

Setting the Button separator style in a menu item's notebook causes the other menu items on the menu bar to disappear.

The disappearing menu items reappear when you cursor through the menu items or change the style back to its original value. The menu item's label displays the default value "Menu item" while the Button separator style is set on.

Message Subfile

AddMsgText and MsgSubText

The message substitution attribute, **MSGSUBTEXT** also works with the **ADDMSGTEXT** attribute. The reference material only mentions the **ADDMSGID** attribute. To use message substitution text with the **ADDMSGTEXT** attribute, specify the message substitution variables as specified by the **ADDMSGID** attribute.

Scroll Bars

For the message subfile and list box parts, vertical scroll bars behave differently for OS/2 and Windows:

- In Windows, the vertical scroll bar appears only where there are sufficient entries.
- In OS/2, the scroll bar always appears, but is only enabled when there are sufficient entries.

Multi-Line Entry

Color

If you drag-and-drop a color onto the scroll bar of a multi-line entry, that color is not persistently saved.

The multi-line entry is changed to the new color, but when you close and re-open the window, the color is changed back to its original color.

The following color attributes are *not* supported in the Windows run time:

- BackMix
- ForeMix
- BackColor
- ForeColor

Wordwrap attribute

In Windows and WinOS2 run times, the Wordwrap attribute does not function if the horizontal scroll style is set.

The WordWrap attribute can only be set at build time.

Push Button

Color attributes in Windows

The following color attributes are *not* supported in the Windows run time:

- BackMix
- ForeMix
- BackColor
- ForeColor

HelpEnable under Windows and Win-OS2

The HelpEnable attribute for the push button and graphic push button part does not work in the WIN-OS/2 or Windows run-time environments. To display the help, give the part focus and press **F1**.

Press event under Windows

In the Windows run time, if the cursor is in an entry field and the Enter key is pressed, the currently highlighted push button generates a PRESS event even if the entry field has focus. You can determine which push button should be activated in this case, by setting the **Highlight** attribute for the push button when the entry field receives the **GOTFOCUS** event.

MouseFocus

The MouseFocus attribute has no effect during Windows run time.

Radio Button

Color attributes in Windows

If the color is changed from its default value, the new color is ignored at run time and the part is displayed with its default colors.

Initially Selected

In the Radio button Notebook settings, you can indicate if the radio button is to be initially selected or not. Note that only one radio button in a group may be selected at one time. If you select more, only the last one in the group is selected.

MouseFocus

The MouseFocus attribute has no effect during Windows run time.

SelectIdx

For the SelectIdx attribute to return the correct radio button index, only radio buttons should be in a group. Adding other part types to the same group may result in erroneous values returned.

Windows run-time restriction

For the **Radio button** part, ensure that only one radio button is set in the builder. When the application is run in Windows, the VRPG run time does not ensure only one radio button is set.

Slider

Settings

If the slider is displayed in an unexpected way after certain settings are changed, press the Reset button to get back to the original setting for the slider.

When resetting the color of a slider part or combo box to the system default, the color is not visibly changed until you close and reopen the window.

Spin Button

The *Client/Server Visual Programming with VRPG Client for OS/2*, and the *VRPG Client for OS/2 Parts Reference Version 3* state that the spin button is not supported for Windows run time.

This is an error. The spin button part **is** supported for the Windows run time.

Static Text

Color attributes in Windows

If the color is changed from its default value, the new color is ignored at run time and the part is displayed with its default colors.

Subfile

Number of subfile fields

The subfile part allows a maximum of 40 fields to be defined.

MultiSelect and ExtSelect

The *Parts Reference* states that **MultiSelect** must be enabled before **ExtSelect** is. This is not true. The **MultiSelect** attribute has no effect on **ExtSelect**. The **ExtSelect** attribute overrides the settings for either **MultiSelect** or single selection.

Repaint attribute

The **Repaint** attribute was only useful in VRPG Release 1. There is no need to use it in later releases.

Timer

TimerTicks

The **TimerTicks** attribute is calculated in the following way:

$\text{TimerTicks} = (\text{Elapsed Time in milliseconds}) / (\text{Interval} * \text{Multiplier})$

Window (and Window with Canvas)

Window with Canvas

In V3R6, the window with canvas part actually consists of two parts: a window frame and a canvas.

Color (BackColor, BackMix)

In Release 1 of VRPG, you set a window's background color by setting the **BackColor** and **BackMix** attributes.

For Release 2, setting these attributes only affects the color of the title bar. To change the interior color of the window, you must set these attributes for the canvas part.

To find the name of the canvas part, look at the tree view of the project in the GUI Designer. The canvas part appears as the child of the window. This is the canvas part whose color attributes you can change.

Owner window

Any value specified for the owner window in the Window's notebook is not supported during run time.

As a result, the ability to move with the owner is lost.

System Modal attribute

When the **System modal** attribute is set for a window part in its settings notebook, that window is modal to the desktop when it is created. Be aware that OS/2 only allows one window at a time to be modal to the desktop. Therefore, if a second window is created that has the System Modal attribute set, that window is modal to the desktop and the first window loses its modality. If the second window is subsequently closed, the first window remains modeless.

Terminate on Close attribute

If you create cascaded application projects, closing one of the cascaded projects closes the entire application, not just the cascaded one. This is because the default state for the **Terminate on Close** attribute for the first window of a project is enabled. VRPG Client treats cascaded projects as DLLs at run time.

Title bar in Windows runtime

Window parts display a title bar even when **Title Bar** is disabled in the settings notebook. When the setting for **Title Bar** is not selected, the user is still able to move the window around using a mouse on the title bar area, but has to close the window using the keyboard.

Title bar font

If you drag-and-drop a font onto the title bar area of a design window, that font is not persistently saved.

The title bar is changed to the new font, but when you close and reopen the window, the font is changed back to its original font.

WindowMode attribute

When a window is restored from a maximized state, the **WindowMode** attribute may indicate that the window is still in the maximized state when it is not.

Event Attributes

Before event attributes can be used in your program, they must be defined on a definition specification (D) with the correct type and length.

The following table lists the event attributes with corresponding data types and lengths:

Event Attribute	Type	Length
%Alt	Numeric	1
%Button	Numeric	1
%Character	String	2
%Control	Numeric	1
%Data	String	*
%ExecuteAck	Numeric	1
%Item	String	*
%MouseX	Numeric	4
%MouseY	Numeric	4
%NewHeight	Numeric	4
%NewWidth	Numeric	4
%PanelItem	Numeric	2
%PokeAck	Numeric	1
%Shift	Numeric	1
%Status	Numeric	1
%Window	String	10

Note:

: * = define length as required.
%Character for KeyPress event, length of 1 may be specified

Example:

```
DName+++++ETDsFrom+++To/L+++IDc.Keywords+++++
D %ExecuteAck      S              1P 0
D %Data            S              32A
```

VRPG Language and Compiler

The following sections contain hints and tips about the VRPG language and the VRPG compiler.

Printer Files

- When creating printer files, make sure the record length matches the width of the printer you intend to print on. If the record length is too long, printed data is truncated.
- If a Skip value that is specified in the PRTCTL data structure is greater than the forms length of the page (FORMSLen keyword), blank pages are suppressed.

Program Termination

Display after termination is initiated

Once an application or component has initiated termination (for example, using the STOP operation code or setting LR to ON), the operation codes SHOWWIN, DSPLY, or SETATR cannot be used. A message dialog is displayed if termination has been initiated with these operation codes.

Application Terminates after DSPLY

If a program does a CLSWIN followed immediately by a DSPLY, the application is terminated if the window being closed is the one that has focus.

Closing the window is happening at the same time as the display of the message box.

The closing window is the owner of the modal message box because it is the one with the focus.

The closing window is closed.

The message box is orphaned, which causes the operating environment to terminate the application.

*PSSR recursion

If you use *PSSR with termination and you do not code the *PSSR to detect recursion, an endless loop results.

Passing Parameters

When passing parameters to an application (the primary component), the parameters are verified against the declarations of the *ENTRY PLIST PARMs and assigned accordingly.

If a parameter cannot be mapped to its corresponding PARM, an error message identifying the incorrect parameter is displayed and the application does not start up.

If the PARMs are externally described as variable length characters or graphic fields, the VRPG compiler treats them as fixed length character fields. In this case, the parameters are assigned values starting at the first byte position without regard to the two-byte length prefix. This may cause problems on output operations.

HIVAL and *LOVAL

*HIVAL and *LOVAL are not supported for the CHAIN, SETLL, SETGT, READPE, and READE operation codes.

If you are writing a client/server application, and you use *HIVAL and *LOVAL with these operation codes, the results may be inconsistent.

To position the cursor in a file, use *START with the SETLL operation code to position the cursor at the beginning of a file. Use *END with SETLL to position the cursor at the end of a file. You can only specify a file name in factor 2 for SETLL because positioning is performed on files regardless of the record format.

When *START and *END are used on a keyed file, the positioning is performed using the keyed access path. For example, *START positions the cursor to the first key in the access path. *END positions the cursor after the last key in the access path.

LPEX Editor

The following sections contain information about the LPEX editor.

Program Specifications

Your VRPG source is maintained as one source file in the directory you specified when the application was created.

When editing existing action subroutines, the LPEX editor locates the action subroutine in the source and scans backwards to include all comments for that action subroutine.

The **Program spec** section initially has an **H** (control specification) following its block comments. If you delete or comment out this **H** spec, the LPEX editor includes these comments with the comments in the first action subroutine.

Subroutines

VRPG extracts subroutines and action subroutines by searching the VRPG source file for matching BEGSR/ENDSR and BEGACT/ENDACT pairs.

If you delete a BEGxx or ENDxx, you do not get a syntax error, but VRPG is unable to correctly extract subroutines.

Full View

When you are editing event logic through the pop-up menu for a part, you are editing an extracted portion of your VRPG source. In this environment, you can make changes to action subroutine names and VRPG keeps track of the Action links.

However, in the **Full View**, VRPG does not manage changes to action subroutine names. If you make changes to subroutine names, you lose the action link to the subroutine. To re-link an event to an action subroutine name that has changed, you must use the **Action Link** dialog to reestablish the link.

Editing Action Subroutines

When editing an action subroutine through a part's pop-up menu, do not add any more action subroutines in the action subroutine you are editing. VRPG does not keep track of these "extra" subroutines.

Writing Online Help

Do not delete the resource identifier (resid=xxx) that VRPG generates when you are creating Help for a part. VRPG uses this value to locate the help for the part. If it is deleted or changed, VRPG cannot find it, and the help for that part is lost.

Part 2. Lab Exercises

This part of the redbook contains hands on Lab exercises that guide you through building an VRPG application.

Starting with building a simple one window application, it leads to extending this program into a complete subfile application accessing the AS/400 database.

After going through this Lab, you have the foundation in VRPG programming that enables you to build your own business applications with VRPG.

Chapter 13. VRPG Client/2 Labs

The objective of the VRPG Client lab is to have the students work with the VRPG Client GUI Designer, create small VRPG Client programs, and successfully build and run them. At the end of the lab, the student should know how to create a simple VRPG application by creating a graphical user interface, writing RPG code to handle the interface events, building the application, and using the debugger to debug the RPG logic.

All of the exercises depend on the previous labs being successfully completed. Ensure that all students have finished the labs and understand the background so they can successfully develop larger VRPG Client applications at home.

The first labs have detailed instructions on how to proceed. This changes when proceeding through the course because it is assumed that students become familiar with the VRPG Client environment and do not require detailed instructions.

It is, however, important that students go through the beginning labs to become familiar with VRPG Client and its terminology.

Note: The pictures in these labs show a similar application being built. Some of the names and icons may be different than the environment you are working with.

Prerequisites

Although not required, familiarity with the RPG language is preferred.

Also, it is helpful if the student is familiar with basic OS/2 operations such as working in the workplace shell and basic mouse operations such as opening folders and performing drag-and-drop operations.

Overall Lab Configuration

The lab should be equipped with OS/2 workstations running OS/2 WARP or later that are connected to an AS/400 system running OS/400 V3R1 or V3R6. The AS/400 connection may be through Communications Manager/2 (CM/2) or Client Access for OS/2.

The AS/400 system must have ADTS installed. If the AS/400 system is at V3R1, PTF SF25794 must be applied. If the AS/400 system is at V3R6, PTF SF26792 must be applied.

The latest version of VRPG Client, 5763VR1, or 5716VR1 must be installed on all workstations.

These workstations must meet the minimum VRPG Client requirements:

- OS/2 Warp or later
- 16MB of memory
- 20MB of swapper
- 486/33 CPU
- CM/2 or Client Access for OS/2

The labs reference data on the AS/400 system. This data is in a library named GUIDES2. Ensure that this library is installed before running the labs.

Overview Exercise Description

It is expected that students know how to deal with a graphical user interface (GUI). They should know how to work with a mouse or similar pointing device. Experience with Windows or OS/2 products, (word processors or spread sheets) is helpful.

Lab 1: Create a Simple VRPG Client/2 Application

Exploring the VRPG Client GUI Designer and Creating Action Subroutines

During this lab, you learn more about the VRPG Client GUI Designer. You walk through the following steps:

1. Exploring the VRPG Client GUI Designer:
 - Starting the VRPG Client GUI Designer
 - Getting familiar with the components of the VRPG Client GUI Designer
 - Using parts and GUI Designer components to create a Graphical User Interface
 - Saving the Graphical User Interface
2. Creating action subroutines:
 - Selecting events
 - Working with the LPEX Editor
 - Creating action subroutines to respond to events
 - Building the application
 - Running the application

What You Should be Able to Do

As a result of this exercise, you can create the first window of your GUI application. In doing so, you can:

- Customize the VRPG Client GUI Designer.
- Create a Window with parts.
- Create action subroutines.
- Save your application.
- Build the application.
- Run the application.

Introduction

You should have a workstation with OS/2 and VRPG Client installed with the minimum configuration previously described. Teams should consist of no more than three members.

USERIDs are given to you by the instructor. However, in this first lab, you are not working on an AS/400 system, so you do not need a USERID for now.

Starting the VRPG Client GUI Designer

To start the VRPG Client GUI Designer for the first time:

1. Locate and double-click on the **ADTS CS for OS/400** icon on the desktop. This opens up the **ADTS CS for OS/400** folder and makes it the active window.
2. Double-click on the **VRPG Client** icon; the **VRPG Client** folder is shown.
3. Double-click on the **GUI Designer** icon in the **VRPG Client** folder to start the **GUI Designer**.

The VRPG Client Product Information dialog is briefly shown followed by a status window indicating the progress in the initialization process of the **VRPG Client GUI Designer**.

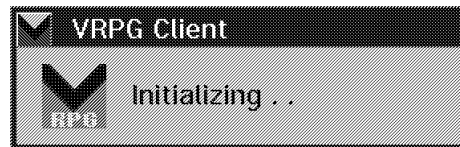


Figure 18. VRPG Client Status Indicator

The GUI Designer, including the **Parts Palette** and the first design window, is shown:

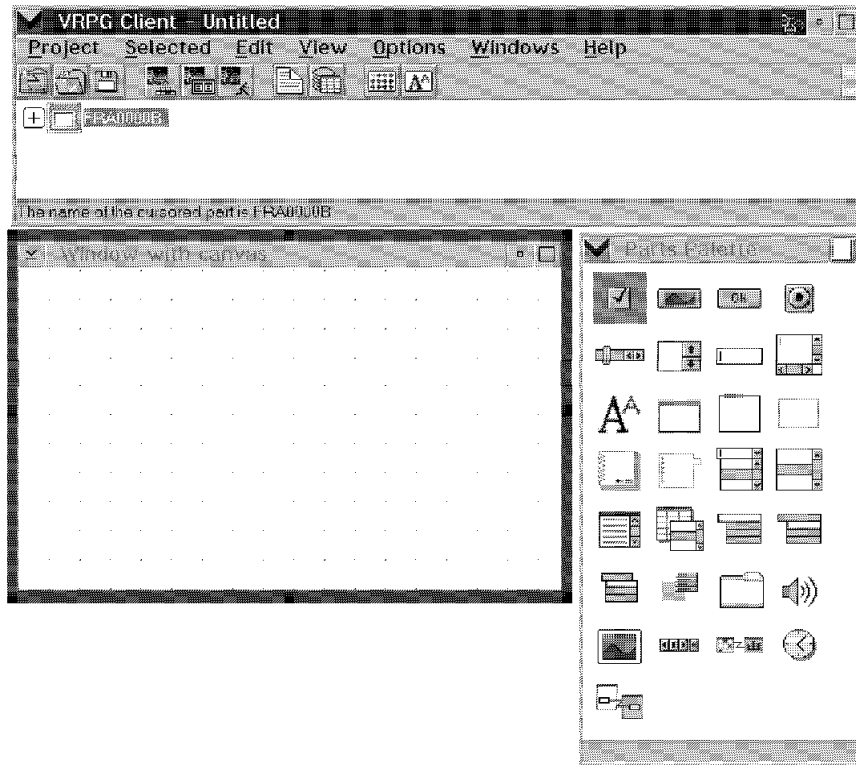


Figure 19. VRPG Client GUI Designer

Rearrange the windows if necessary so they are all visible. Re-size the **Parts Palette** so that all parts are visible.

Components of the VRPG Client GUI Designer

The VRPG Client GUI Designer consists of these major components:

- Menu Bar
- Tool Bar
- Project View
- Parts Palette

The Menu Bar

The **Menu Bar** is located below the title bar of the **VRPG Client** window. It provides a variety of tasks that you can use to create and modify your GUI application, customize the VRPG Client GUI Designer, and get online help.

1. Press F10 to highlight the leftmost menu bar choice.
2. Press the right arrow key to scroll through the choices of the menu bar. While doing this, look at the bottom of the **VRPG Client** window. This is the **Information Area** that gives you a short description of the action that is performed when selecting one of the menu choices (the items contained in the pull-down menu of a menu bar choice).

If you need a detailed description for a specific choice, press F1 while the desired menu bar item is highlighted. A pull-down menu with menu choices is opened, while the information area gives you a short description of the action performed when the highlighted menu choice gets selected.

The Up and Down arrow keys can be used to navigate the menu choices. Detailed help for each choice is available by pressing F1.

To become familiar with the available menu choices:

1. Press F10 to highlight the **Project** menu bar choice.
2. Use the Up, Down, Left, and Right arrow keys to scroll through all the menu choices. Look at the information area at the bottom of the **VRPG Client** window to see what action the choices perform.

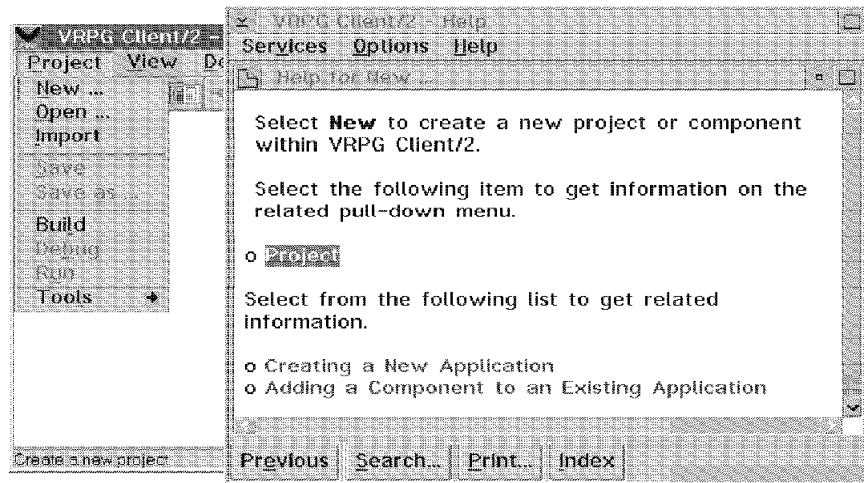


Figure 20. Displaying Menu Help

The Tool Bar

The **Tool Bar** is a menu of icons. It provides a fast path access to some of the menu items on the menu bar. Invoking the action from an icon on the tool bar allows for quicker access to common used menu choices.

1. Move the mouse arrow over the icons of the Tool Bar. A short description of the icons function is displayed in the information area.
2. Click on the **Font Palette** icon. This is one of the rightmost icons on the Tool Bar. The **Font Palette** window is shown.
3. Double-click on the **System** icon (at the upper left corner) of the **Font Palette** window to close it.

The Project View

The **Project View** is used to hold all of the parts you create during your VRPG Client GUI Designer session. The Project View can be changed to display the parts of a project in different views. This is demonstrated in one of the labs.

The Parts Palette

The **Parts Palette** contains parts that you use to create the screen layout for your GUI application. These parts act as a template to create parts from.

1. Move the mouse pointer over the parts on the **Parts Palette**. Notice the information area at the bottom of the Parts Palette. It displays a short description of the part type. To see a detailed description of a part on the **Parts Palette**, select a part with the left mouse button, and press F1.

2. Click on the icon in the upper right corner of the Parts Palette. The Parts Catalog is shown. The Parts Catalog contains all the parts of the VRPG Client GUI Designer categorized by their function, as indicated by the tabs. Click on the icon in the upper right corner of the Parts Catalog to return to the Parts Palette.

Windows or OS/2?

With VRPG Client, you can build an application that runs under Windows 3.1, OS/2, or both. To indicate which program type you are going to create, you need to set the project type. To do this, choose **Project** and then **Set project type...** from the GUI Designer menu bar. The **VRPG Client - Set Project Type** dialog appears. Since we are building an OS/2 application, and the default for this dialog is OS/2, just press the OK or Cancel push button to return to the GUI Designer.

Tip

You can always tell the project type by looking at the icons next to the minimize icon in the GUI Designer title bar.

Creating a Graphical User Interface

During this exercise, you learn how to create a Graphical User Interface by creating windows and adding parts to windows. The Graphical User Interface you are creating now is the first window of a Customer Inquiry application. It is used to prompt for a customer number.

When the GUI Designer is started for a new project, it creates the first design window for you. Notice also that a **Window** part has been added to the Project View.

We use this window as the first window in our Customer Inquiry application. As with all parts, the initial part name is generated by the GUI Designer, with the first few characters indicating the part type (for example, **FRA** indicates a Frame Window part).

A Note About Notebooks

As mentioned earlier, the Parts Palette contains parts that act as templates. Once the part has been created in the Project View, its attributes, such as the part name, can be changed. These changes are made by invoking the parts **Settings Notebook**. A Settings Notebook consists of **pages** as indicated by the labeled **Tabs**. You go to a Settings Notebook page by clicking with the left mouse button on its Tab. To close a Settings Notebook, double-click on its **System Icon** in the upper left corner.

Let's change some of the settings for the initial design window:

1. Double-click on the **title bar** of the design window. The window part **Settings Notebook** is shown.

Tip

A **Window with canvas** part is made up of two parts: the Window, which consists of the title bar and frame, and the Canvas, which is the white area enclosed by the window. Therefore, if you double-click on the Canvas, you get the Settings Notebook for the **Canvas** part.

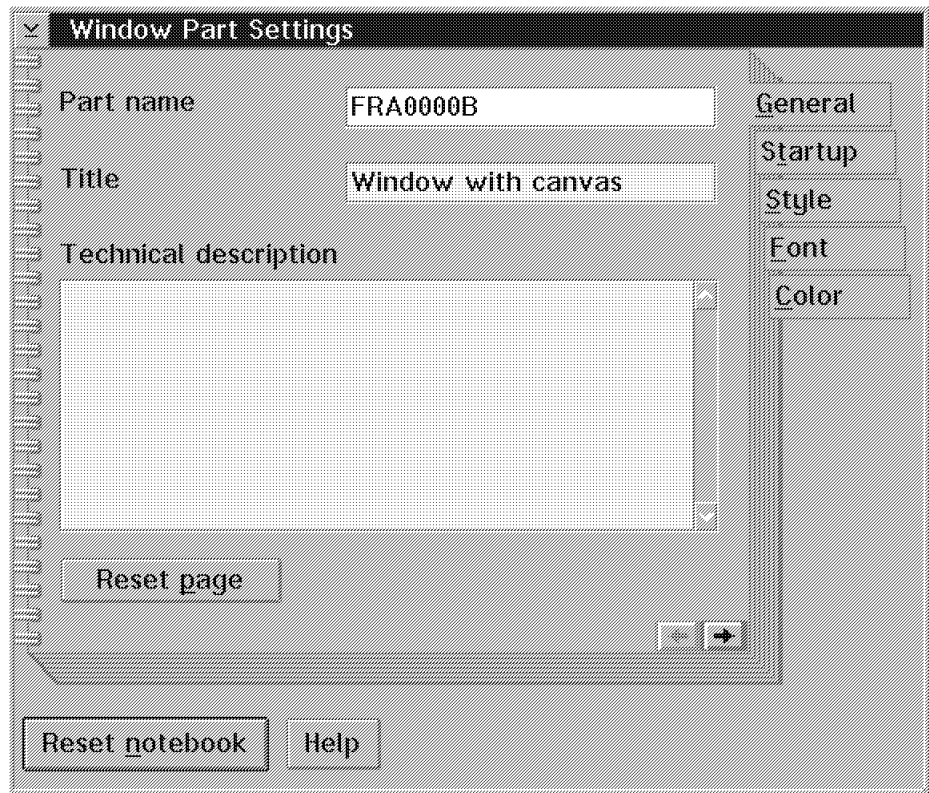


Figure 21. Changing the Window Part Name and Title

2. Change the **Part name** setting to **CUSTINQ** and the **Title** to **Customer Inquiry**.

Note: Although you can type the part name in upper or lower case, it is always folded to upper case by the GUI Designer.

3. Click on the **Style** tab of the Settings Notebook. The **Style** page is shown.
4. Deselect the **Minimize button** and **Maximize button** check boxes.
5. Select **Thick** for the style of the window border.
6. Double-click on the **System** icon of the **Window part** Settings Notebook to close it.

Notice the changes to the window. The new window title is displayed, the border style has changed, and **Minimize** and **Maximize** icons no longer appear on the title bar.

Tip

To change the label and default text of parts, you can also press the **Alt+left mouse button** instead of using the Settings Notebook. When you have made your change, click the left mouse button again. This is called direct editing.

The next step is to add a **Static Text** part to the window to prompt for the customer number to be entered:

1. Find the **Static Text** part in the **Parts Palette**.

Tip

As you move the cursor over the icons on the Parts Palette, a short description of the part type is shown in the Parts Palette information area.

2. Drag-and-drop a Static Text on to the **Customer Inquiry** design window by clicking on it with the right mouse button, dragging the part to where you want it on the design window with the mouse and releasing the right mouse button.
3. Invoke the Settings Notebook for this part by double-clicking on it (the text **Static Text**).
4. Change the **Text** setting on the **General** page to **Enter Customer Number**.
5. Double-click on the **System** icon of the Static Text parts **Settings Notebook** to close it.

You might notice that not all of the changed text is displayed. This is because of the default size setting of the **Static Text** part. To change the size:

1. Click on the **Static Text** part with the left mouse button to select it. The borders of the part are marked by placing small squares, called sizing handles, around it.
2. Move the mouse pointer to the middle of the rightmost three sizing handles. The pointer changes to a double-arrow (pointing to the left and right).
3. Click and hold the right mouse button.
4. Move the mouse pointer to the right until the entire text of the **Static Text** part is visible.
5. Release the right mouse button.

To add the Entry Field that receives the customer number:

1. Find the **Entry Field** part on the **Parts Palette**.
2. Drag an Entry Field part to the **Customer Inquiry** window and drop it to the right of the **Static Text** part.
3. Invoke the **Settings Notebook** for this part by double-clicking on it.

Tip

You can also open the Settings Notebook for a part by selecting the **Open settings** choice from the Pop-up menu for the part, or by selecting **Selected** and **Settings** from the GUI Designer menu bar when the part is selected.

4. Change the **Part name** setting on the **General** page to **CUSTNO**. This is the variable name the customer number can be accessed by later within VRPG Client code.
5. Click on the **Data** tab to get to the **Data** page.
6. Change the length to **7**.
7. Go to the **Validation** page.
8. Select the **Range** radio button.
9. Enter **0000000** as the Minimum value and **9999999** as the Maximum value.

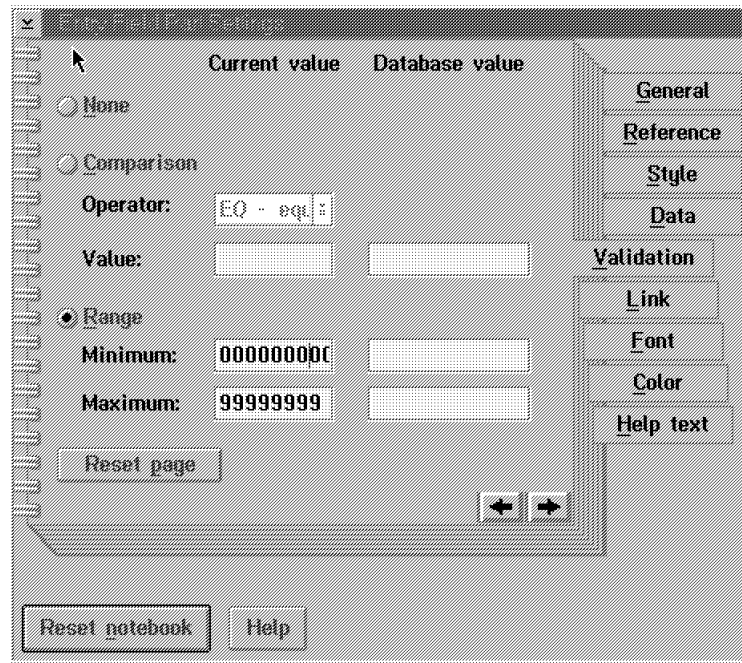


Figure 22. Icon View of Parts

10. Double-click on the **System** icon of the Entry Field part Settings Notebook to close it.

Probably the Static Text and Entry Field parts are not aligned with each other. To do this:

1. Move the mouse pointer above and to the left of the **Static Text** part.
2. Click and hold the left mouse button.
3. Drag the mouse down and to the right so that the two parts are covered by a gray rectangle. This gray rectangle is referred to as a **Selection Rectangle**.

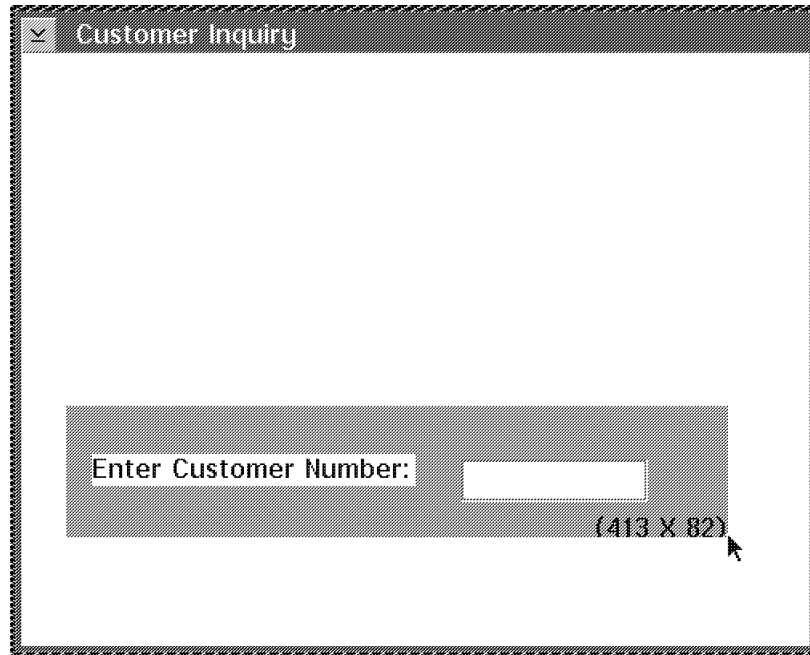


Figure 23. Selecting Parts

4. Release the left mouse button. The two selected parts are highlighted with a dark border.
5. Move the mouse pointer to the **Static Text** part.
6. Click the right mouse button. The Pop-up menu for this part is shown.

Tip

When more than one part is selected, invoking the Pop-up menu for one of the selected parts makes that part the **Anchor** part. Any alignment operations are relative to that part.

7. Choose the **Align...** choice on the Pop-up menu. A sub-menu is shown.
8. Choose the second alignment choice offered. As depicted by the choice icon, this aligns the two parts to an imaginary horizontal axis running through the **Static Text** part.

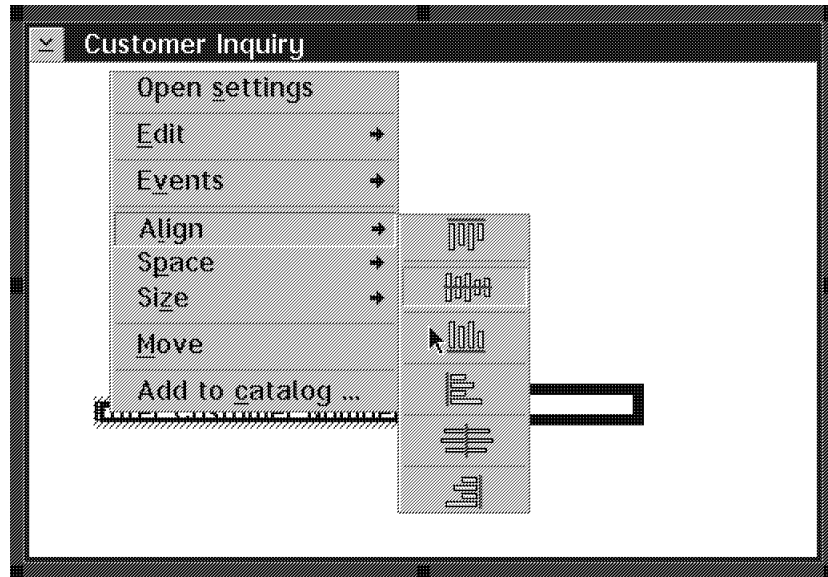


Figure 24. Alignment Tools

To complete this window, we need to add two push buttons, an **OK** push button to process the customer number and an **Exit** push button to end the application. To add these push buttons:

1. Find the **Push button** part on the **Parts Palette**.
2. Drag a push button part to the **Customer Inquiry** window and drop it below the **Static Text** part.
3. Invoke the **Settings Notebook** for the push button.
4. Change the **Part name** setting on the **General** page to **PSBOK** and the **Text** setting to **OK**.
5. Go to the **Style** page.
6. Select the check box for the **Default** option. This makes it the part that gets selected when the Enter key is pressed.
7. Double-click on the **System** icon of the push button part Settings Notebook to close it.

To create the second (Exit) push button, you create a duplicate of the OK push button:

1. Move the mouse pointer to the OK **Push button** part and click the right mouse button to get the Pop-up menu for this part.
2. Choose the **Edit** menu choice and the **Duplicate** sub-menu choice. A second OK push button is created.
3. Move the mouse pointer to this new **Push button** part.
4. Press and hold the right mouse button and drag the **Push button** part below the **Entry Field** part.
5. Release the right mouse button.

To change the text of the push button:

1. Press and hold the **ALT** key and select the push button part with the left mouse button. The text of the **Push button** part becomes editable.

2. Type in **Exit**.
3. Deselect the part by pressing the left mouse button while the mouse pointer points outside the **Push button** part.

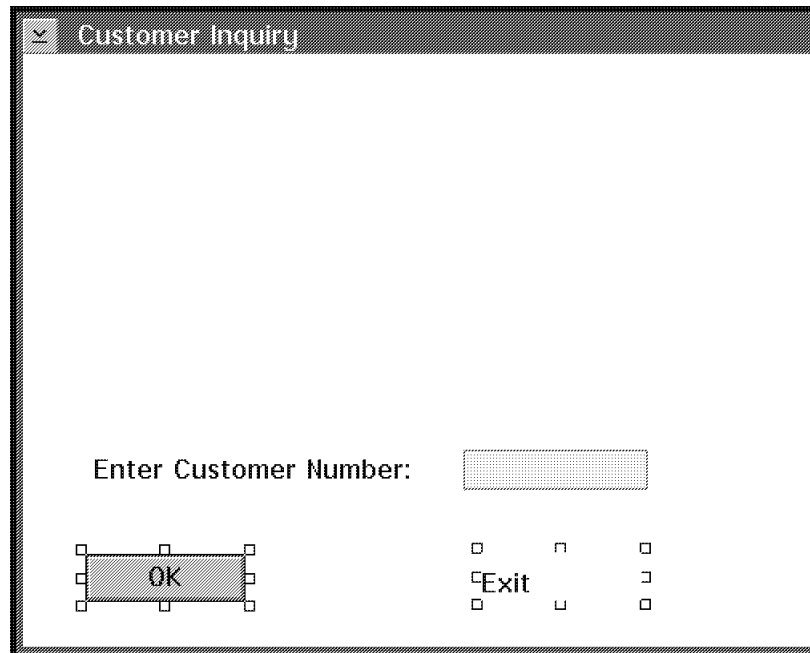


Figure 25. Changing Push Button Label

Because this push button part has been created as a duplicate of another push button, it also adopted the same attributes with the exception of the part name which is generated by the GUI Designer to avoid name conflicts in this window. To change the name for the Exit **Push button** part:

1. Invoke the Settings Notebook for the **Exit** push button part by double-clicking on it.
2. Change the **Part name** setting on the **General** page to **PSBEXIT**.
Note: The text is already changed to **Exit**.
3. Go to the **Style** page.
4. Deselect the check box for the **Default** option.
5. Double-click on the System icon of the push button part Settings Notebook to close it.

Now, the two **Push button** parts need to be aligned. The left border of the OK **Push button** part must be aligned to the left border of the **Static Text** part, while the right border of the Exit **Push button** part needs to be aligned to the right border of the **Entry Field** part. To do the alignment:

1. Move the mouse pointer above and to the left of the **Static Text** part.
2. Press and hold the left mouse button.
3. Drag the mouse so that the **Static Text** part and the OK **Push button** part are covered by the Selection Rectangle.
4. Release the left mouse button. The two selected parts are highlighted.

5. Move the mouse pointer to the **Static Text** part, click the right mouse button to get the Pop-up menu, and choose the **Align...** menu choice from the Pop-up menu.
6. Choose the fourth choice offered. This aligns the OK **Push button** part to the left border of the **Static Text** part.

To align the Exit **Push button** with the Entry Field:

1. Move the mouse pointer above and to the left of the **Entry Field** part.
2. Press and hold the left mouse button.
3. Drag the mouse so that the **Entry Field** part and the Exit **Push button** part are covered by the Selection Rectangle.
4. Release the left mouse button. The two selected parts are highlighted.
5. Move the mouse pointer to the **Entry Field** part, click the right mouse button to get the Pop-up menu, and choose the **Align...** menu choice from the Pop-up menu.
6. Choose the sixth choice offered. This aligns the Exit **Push button** part to the right border of the **Entry Field** part.

Now, align the two **Push button** parts to each other horizontally in the same way you aligned the **Static Text** and **Entry Field** parts.

Finally, the size of the window has to be changed to contain the parts in it:

1. Move the mouse pointer over the upper right corner of the window. It becomes a double-arrow.
2. Press and hold the right mouse button.
3. Drag the mouse to re-size the window until all of the parts are in the middle of the window.

Note: None of the parts created within the **Customer Inquiry** window are moved when the window is re-sized. This is because parts remain relative to the left and bottom edges of the **Window** part. You need to be aware of this since some re-sizing operations move parts so they are not longer visible.

The resulting window should look similar to the following:

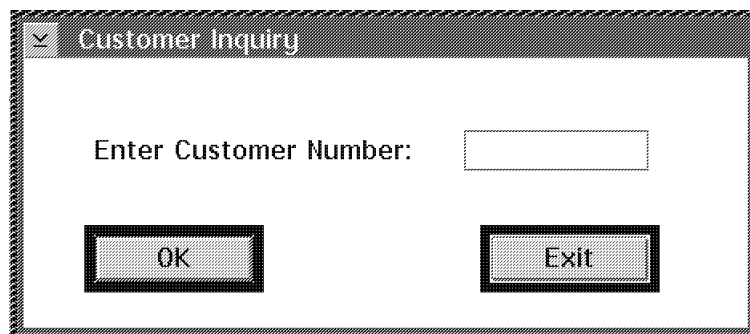


Figure 26. Completed Customer Inquiry Window

Saving Your Project

Now that you have finished your first Graphical User Interface, you should save your project.

1. Move the mouse pointer to the **Save Project** icon on the Tool Bar.

Tip

As you move the mouse pointer over the Tool Bar, the information area is updated to indicate the function of each button.

2. Click on this icon and save your project. Since this is a new project, and has not yet been named (notice the title bar on the GUI Designer says **Untitled.**), the **VRPG Client Save as Application** window is displayed. This window is used to name your project, indicate in which folder it should be saved, and where the project files should be saved.

Make the following entries on this window:

- In the **Application name** entry field, type a title for your application (for example, *Customer Inquiry*).
- Leave the folders selection at **Desktop**.
- In the **Source file** entry field, type **GuiDes2**.
- Replace the value of the **Source directory** entry field with **x:\GUIDES2**.

Note: Replace **x** with a valid drive letter on your workstation.

- Press the **OK** push button to save your project. Notice that after the project has been saved, the GUI Designer title bar is updated to show the project name.

This ends Part 1 of this lab. You use this project in the next part of this lab so please do not exit the GUI Designer at this time.

Creating Action Subroutines with VRPG Client

You use the window that you created in the previous exercise. You are adding an **Action Subroutine** for the Press event of each push button. Recall that an Action Subroutine is the VRPG Client logic that is invoked to handle an **Event** on the Graphical User Interface.

Creating an Action Subroutine

Now that you have finished designing this window, you can add some code to give it some function. In this exercise, you create the Action Subroutine for the Exit push button.

1. If the CUSTINQ design window is not open, open it by double-clicking on its icon in the GUI Designer Project View.

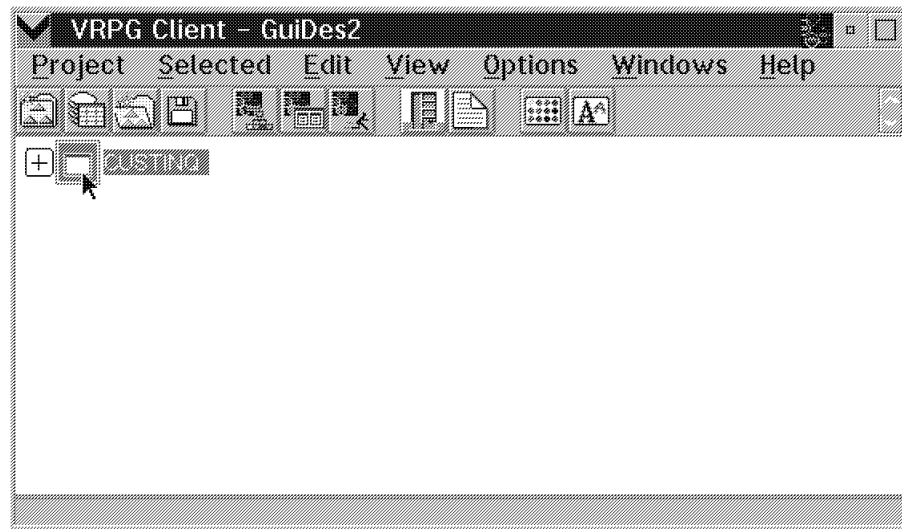


Figure 27. Opening a Window in the Project View

2. Select the **Exit** push button using left mouse button. Click the right mouse button to get the Pop-up menu. Choose **Events** and choose **Press** from the sub-menu.

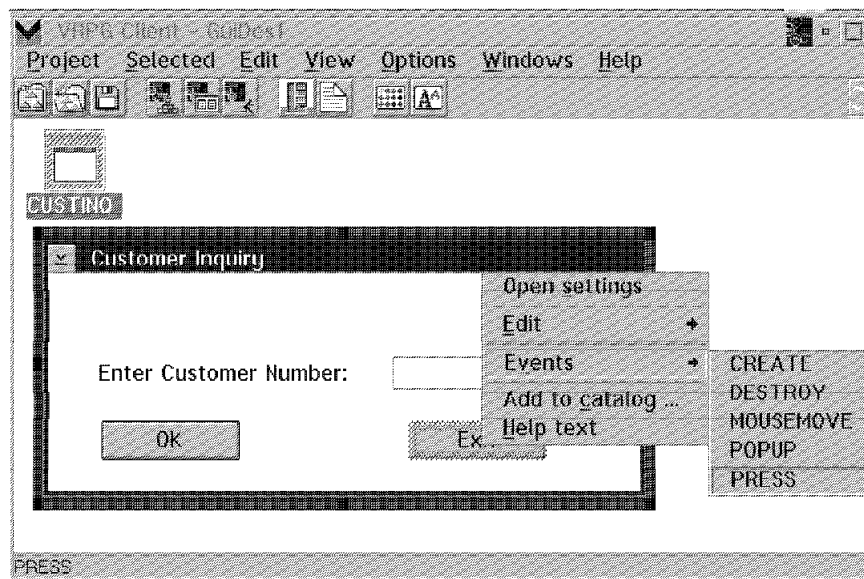


Figure 28. Selecting an Event

3. This starts the LPEX Editor.



Although the framework for the RPG code has been built, the application logic is not there. It must be added. In this example, the code to terminate the application must be added.

4. Position the mouse pointer anywhere on the line with the BEGACT operation code and click left mouse button. This positions the cursor.
5. Choose the **Edit** option from the LPEX Editor menu bar and choose **Insert with Prompt** from the pull-down menu.

You can also perform an Insert with Prompt by pressing **Shift+F4**.



- 118**
- Installing, Setting Up and Working with VRPG 118 CODE/400

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+.
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
*
C                MOVE      *ON          *INLR
```

Figure 31. Code to Terminate the Program

7. There is extensive online help available for the VRPG Client language. Before leaving this prompt window, position the cursor back into the **Operation and Extender** field. Press F1 (HELP). A help window for the current value of the Operation code field is displayed. In this case, help for the **MOVE** operation code is shown. To close the Help window, double-click on the icon at the top left corner of the window.
8. Press the **OK** push button to add this line of code to the source.
9. To end prompting, press the Cancel push button on the Prompt window.

Tip

Most dialogs that have a cancel push button can also be closed by pressing the **Esc** key.

You notice that this line of code was added directly below the BEGACT statement.

10. To save this change, choose the **File** option from the LPEX Editor menu bar and choose **Save**.
11. Close the LPEX Editor by double-clicking on its system menu.

Let's re-cap what an Action Subroutine is:

In the Event-driven method of creating applications, what happens in the program is controlled by the **Events** occurring on the GUI. An Event can be linked to an Action Subroutine. Action Subroutines are easily found in the code when looking at the complete VRPG Client source. An Action Subroutine begins with the BEGACT operation code and ends with an ENDACT operation code. The logic between these two statements is the business logic that is executed when the event is generated.

Now, you can create an Action Subroutine for the OK push button.

Creating an Action Subroutine for the OK Push Button

You can add an Action Subroutine that changes the color of this push button when it is pressed. If the color of the push button is **Red**, it is changed to **Gray**; otherwise, it is set to **Red**.

This might not seem to be the most useful code but it gives you the opportunity to experiment with some of the new operation codes, namely:

SETATR: (Set attribute)

GETATR: (Get attribute)

Later on, you can add some more meaningful logic for the **Press** event of this push button.

1. Select the **OK** push button with the left mouse button. Use the right mouse button to get the Pop-up menu. Choose **Events** from the Pop-up menu and choose **Press**, which invokes the LPEX Editor.

2. You want to write logic to get the background color of this push button. You do this by using the **GETATR** operation code. Your logic determines the new color to set it to; use the **SETATR** operation code to change it.
3. This logic sets the background color to red if it is not currently red; otherwise, it sets the background color to gray. Your logic should look similar to the following figure.

Tip

VRPG Client statements can be entered in upper, lower, or mixed case including part names and attribute names.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
CLON01Factor1++++++Opcode(E)+Factor2++++++Result++++++Lent++D+HiLoEq
*
C      PSBOK      BEGACT   PRESS      CUSTINQ
C      'PSBOK'    Getatr   'BackColor' BCOLOR      2 0
C                      If     BCOLOR <> *RED
C      'PSBOK'    Setatr   *RED      'BackColor'
C                      Else
C      'PSBOK'    Setatr   *PALEGRAY 'BackColor'
C                      EndIf
C                      ENDACT
```

Figure 32. Code to Terminate the Program

Tip

Some basic LPEX Editor commands:

- **ALT+L** marks a line.
- **ALT+U** unmarks a line.
- **ALT+D** deletes a marked line.
- **ALT+C** copies a marked line.
- **ALT+M** moves a marked line.

Press Enter to insert a new line.

RPGIV: Base for VRPG Client Language Definition

The VRPG Client language is based on the RPGIV language syntax. Both upper and lower case letters are allowed in the source code. This greatly enhances the readability of the code, as well as the ease of coding. The second RPGIV feature that is notable in the VRPG Client is the 10 character names. Not only does this make the code more legible and the names more meaningful, it also means that the names can be the same as the names used in other AS/400 languages as well as names defined in Data Description Specifications (DDS) without the need to rename them.

Some LPEX Editor Features

Now that you have added the second Action Subroutine, let's look at some of the features of the LPEX Editor so you can later easily find your way around and utilize them.

The LPEX Editor Tool Bar

This version of the LPEX Editor also has a tool bar to access the most commonly used LPEX Editor functions. By default, the Tool Bar, is not displayed. To use the Tool Bar:

1. If an LPEX Editor window is not currently open, open one. You can open an LPEX Editor window by choosing **Project** and **Edit source code** from the GUI Designer menu bar.
2. Choose **View** from the LPEX Editor menu bar and choose **Tool Bar**.
3. The LPEX Editor Tool Bar is displayed. Note as you move the mouse pointer over the Tool Bar icons, the LPEX Editor information area is updated to indicate the function of the item.
4. To remove the Tool Bar, choose **View** and choose **Tool Bar**.

Sequence Numbers

If you prefer, the LPEX Editor can display a prefix area in the source view. This prefix area is similar to the one in SEU. It contains the statement number as well as supporting line commands. To enable the sequence numbers:

1. If an LPEX Editor window is not currently open, open one.
2. From the LPEX Editor menu bar, choose **View** and **Sequence numbers**.
3. The sequence numbers are displayed in the prefix area.
4. To execute a prefix area command, type the commands in the prefix area and press **ALT+Enter**.
5. To remove the prefix area, choose **View** and **Sequence numbers**.

Token Highlighting

The LPEX Editor highlights the **Tokens** (specification fields) of your VRPG Client program source, providing a separate color for each improving readability.

When you make changes to a line, the token colors are updated only after you move your cursor off the line.

To see how token highlighting works:

1. Move the cursor to the first Calculation specification in the VRPG Client source.
2. Position the cursor to column 7 (right next to the 'C'). Refer to Figure 33 on page 122.

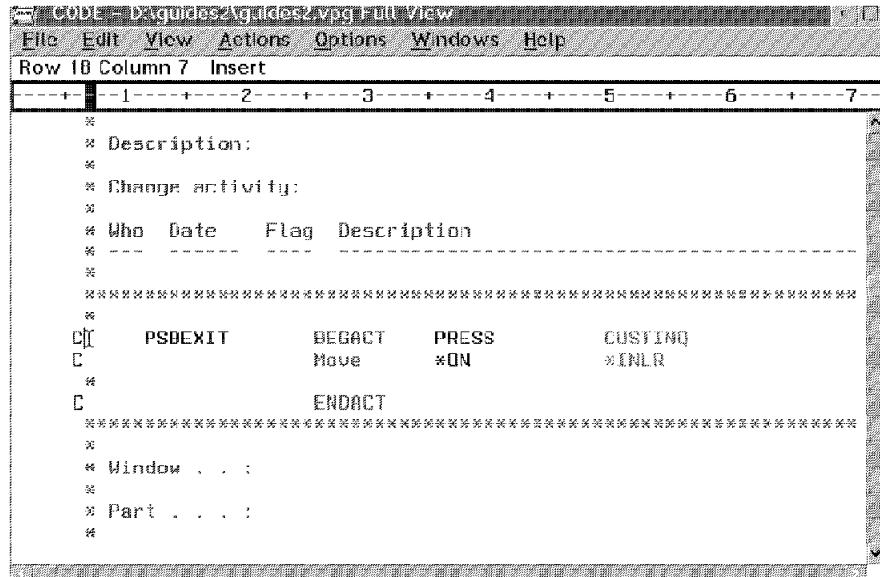


Figure 33. Positioning the Cursor to Column 7

3. Type an asterisk (*).
4. Move the cursor off the line and watch what happens. The line where the "*" was typed has become a comment line and its color changes accordingly.
5. Move the cursor back to column 7 and remove the "*". Move the cursor off that line of code.
6. The token highlighting changes to reflect that this is a non-commented "C" specification.

Displaying Types of Lines

Using the LPEX Editor, it is possible to have only particular types of source lines displayed at a time. To do this:

1. Choose **View** from the LPEX Editor menu bar. The options available on the pull-down menu lists the types of line selections that can be made.
2. Choose **Comments**. The LPEX Editor now contains only those VRPG Client statements that are comments (see Figure 34 on page 123).

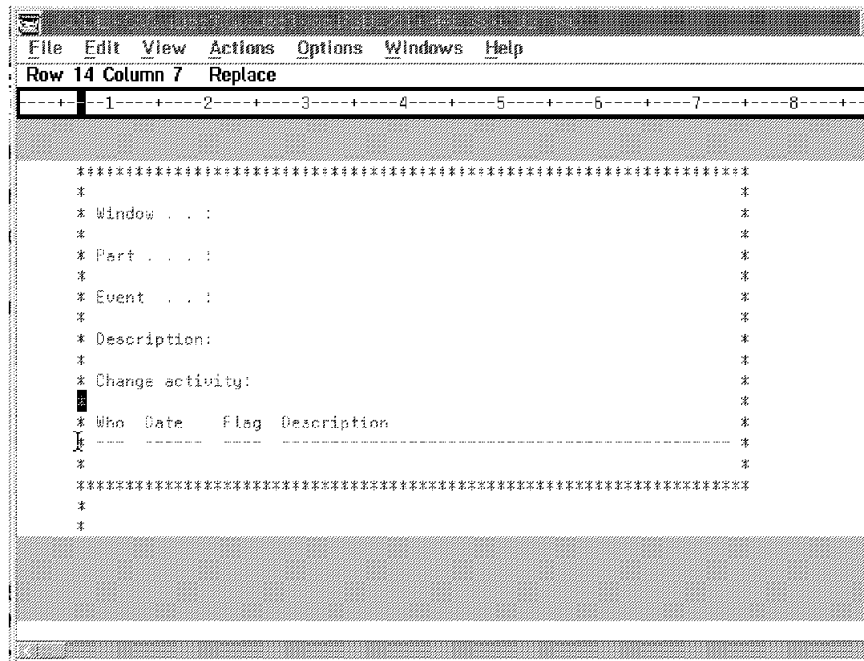


Figure 34. Only Comment Lines are Shown

3. Choose **View** from the LPEX Editor menu bar and choose **Include all**. All statement types are displayed.

In addition, the choices in the **View** pull-down can be used to include only control specifications, user subroutines, and Action Subroutines. The **Include** choice allows the selection of only those lines containing a particular character string. As the LPEX Editor is fully-programmable, it is possible to create macros to include/exclude all types of source statements based on specific criteria.

Syntax Checking

Syntax checking is available for VRPG Client code, and by default, is active. The syntax of the VRPG Client code is checked automatically when a change is made to a line and the cursor is moved off the line.

When errors are found, they are displayed in the message area at the bottom of the LPEX Editor.

To see how syntax checking works:

1. Move the cursor to the statement containing the **MOVE** operation code.

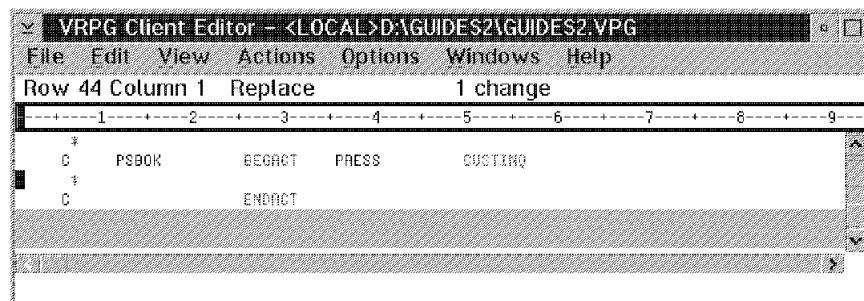


Figure 35. Row and Column Information

2. Make sure the LPEX Editor is in Replace mode (the mode indicator is directly to the right of the row and column information). Use the **Insert** key to change modes.
3. Create an error by removing the "E" from the MOVE operation code (see Figure 36).

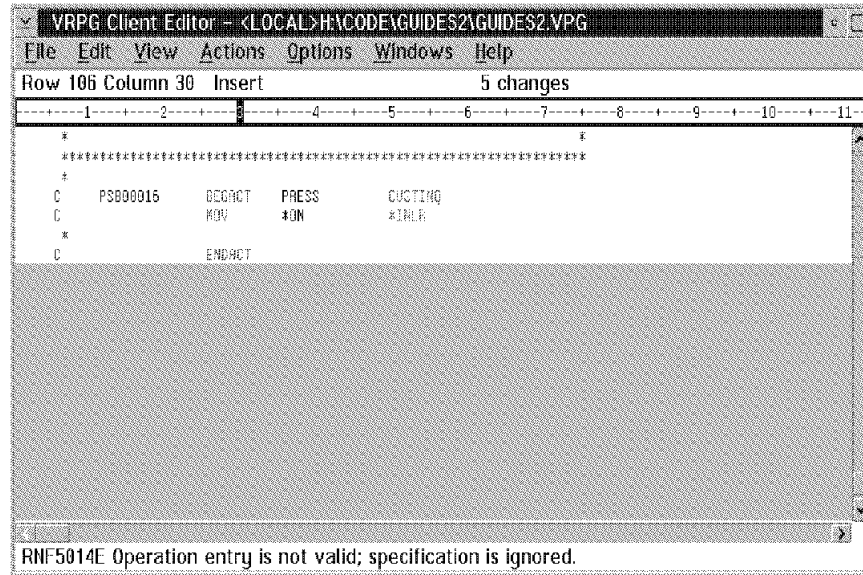


Figure 36. Creating a Syntax Error

4. Move the cursor off of this statement. An error message is displayed in the message area at the bottom of the LPEX Editor window showing the syntax error (see Figure 36).
5. Correct the error by typing the character "E" back into the operation code "MOVE".
6. Move the cursor to the next line. The message area is cleared.

Using Format Lines

The format line is at the top of the LPEX Editor window, just below the menu bar and status line. A format line is used to help keep track of the columns in a particular specification line. The content of the format line can vary to reflect the particular type of specification being keyed (for example: F specs, C specs, D specs, and so on).

To display a format line:

1. Position the cursor to any uncommented Calculation specification line by clicking on the line with the left mouse button, or by using the arrow keys and clicking the left mouse button.
2. Choose the **View** menu bar choice and choose **Refresh format line**.
3. The format line changes to reflect an VRPG Client Calculation specification since the cursor was on a C-specification when the format line was requested.

Tip

The format line can also be refreshed by pressing **Ctrl+R** with the cursor on a statement.

4. You can move the cursor right or left with the arrow keys to go from character-to-character, or with the Tab key to go from field-to-field. An indicator on the format line moves with the cursor to show in which column the cursor is positioned. Try moving the cursor and watching the indicator on the format line.
5. Move the cursor to a comment line (an asterisk in column 7).
6. Choose the **View** menu bar choice and choose **Refresh format line** (or press Ctrl+R). A format line for comment specifications is shown since the cursor was on that type of specification when you requested a refresh.
7. To select a format line for any specification you want, choose the **View** menu bar choice and choose **Select format line**. The **VRPG Client Format Line Selection** window is shown.
8. Select the **C-Calculatation** radio button and press the **OK** push button. The format line changes to reflect a calculation specification.

Now that you have created the push button logic and explored some of the features of the LPEX Editor, save your work by choosing **File** and **Save** from the LPEX Editor menu bar. Close the LPEX Editor by double-clicking on its system menu.

The next step is to build your application.

First close the design window (the Customer Inquiry window) by positioning the mouse pointer anywhere on the its window title bar and click with the right mouse button. From the Pop-up, choose **Close**.

Tip

You can also close a design window by double-clicking on its system menu.

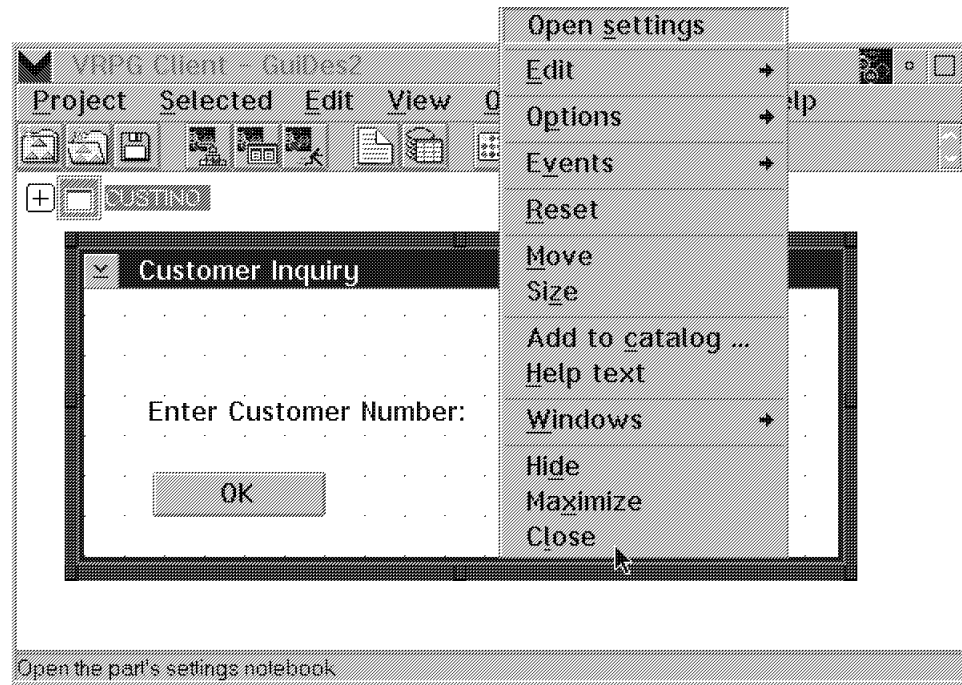


Figure 37. Closing the Design Window

Building the Application

Before the application can be run, it must be built. Before starting the build, let's have a look at the build options dialog that allows you to specify the options used to build this project.

1. Choose **Project** and **Build options** from the GUI Designer menu bar. Since we are building an OS/2 application, choose **OS/2** from the sub-menu.
2. The **VRPG Client Build Options** dialog is shown.

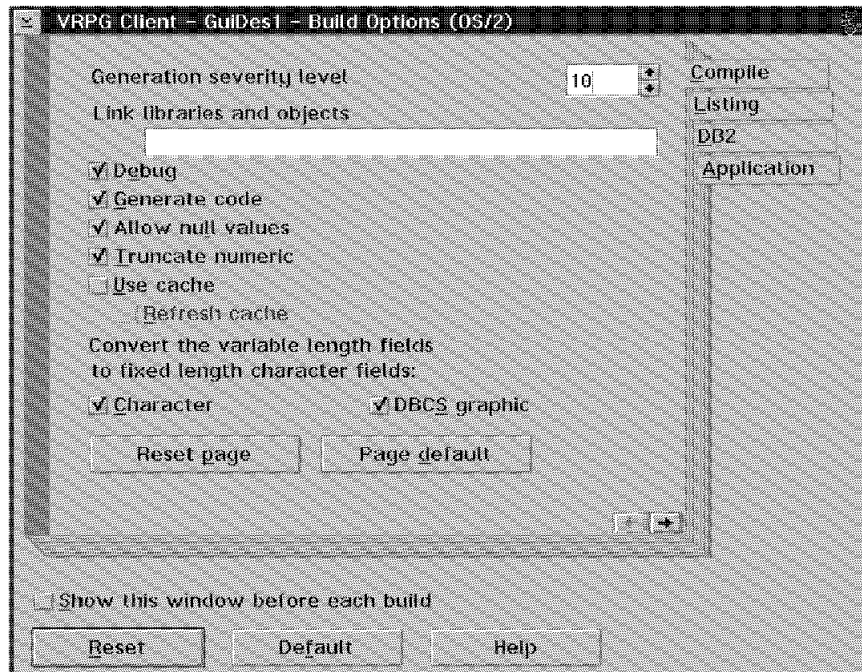


Figure 38. Build Options Dialog

3. Notice the check box at the bottom of this dialog, **Show this window before each build**. If this check box is selected, the build options dialog is displayed each time a project build is performed.
4. Go to the **Application** page of the build options dialog.
5. The icon shown on this page is the icon that represents the application executable on the desktop if the build is successful. For this lab, we use the default icon.
6. Close the build options dialog.

Now, let's build the project:

1. Choose **Project** from the GUI Designer menu bar and choose **Build**. A sub-menu is shown that indicates if an OS/2 program or a Windows program is to be built. Notice, that since we are building an OS/2 application, the **Windows** menu item is disabled (grayed out). Choose **OS/2**.
2. The **VRPG Client - Save project** window appears. Press the **Save** push button to save your project.
3. A status window is now shown to indicate that the project is being built. When the build is complete, another window is shown to indicate if the build is successful or not.

Running the Application

If the build is successful, you are now ready to run the application. There are two ways to do this:

1. From the **Project** pull-down menu, choose **Run** and **OS/2**.
2. Press the **Run OS/2 version of the project** Tool Bar icon.

The window you designed is displayed. Press the **OK** push button and check whether your logic is correct and the background color of the push button changes.

To terminate the program, press the **Exit** push button.

This concludes the first lab.

End the GUI Designer by double-clicking on its system icon.

Lab 2: Exploring VRPG Client GUI Designer and Accessing AS/400 Databases

During this exercise, you work on the application that you created in the previous lab. The following tasks are covered in this lab:

- Starting the VRPG Client GUI Designer for an existing application
- Customizing the VRPG Client GUI Designer
- Creating a **User-Defined Part** and adding it to the **Parts Palette** and **Parts Catalog**
- Creating an instance of a User-Defined Part
- Referencing fields in a DB2/400 database
- Performing more **Alignment**, **Sizing**, and **Spacing**

As a result of this exercise, you create a window that is used to display customer information given the input from the window you created in the previous lab.

What You Should be Able to Do

As a result of this exercise, you can customize the VRPG Client GUI Designer to suit your needs.

Also, you are able to access an AS/400 system to:

- Use existing DB2/400 field definitions.
- Use externally described AS/400 files in your VRPG Client program.
- Read data from the AS/400 system.
- Read and write data from/to a window.
- Build a VRPG Client application accessing AS/400 data.
- Run a VRPG Client application accessing AS/400 data.

Starting VRPG Client GUI Designer for Existing Project

When a VRPG Client project has been saved, an icon that represents it is shown on the desktop with the text that was specified when the project was first saved.

Note: The icon is shown on the desktop if that is the location you chose on the **Save as** dialog as is the case in the lab. It is possible to save a project in another folder, in which case, the icon is in that folder.

There are two methods of opening a VRPG Client project. The first involves dragging-and-dropping the project on the VRPG Client GUI Designer icon. This

method is demonstrated by the instructor during the lab. For this lab, we use the second method, opening a project from its pop-up menu:

1. Locate the your project icon on the desktop.
2. Invoke its pop-menu by clicking the right mouse button.
3. From the pop-up menu, choose **Edit**.
4. The VRPG Client initialization window is displayed while GUI Designer opens the project.

Customizing VRPG Client GUI Designer Components

Many of the VRPG Client GUI Designer components can be customized to suit your personal preferences. In the following exercises, you perform some customization.

Customizing the Project View

As mentioned earlier, the Project View presents a view of the parts that make up your project. By default, the Project View is in **Tree View**. To change the view, follow these steps:

1. Choose **View** from the GUI Designer menu bar.
2. Choose **Icons**. The Project View changes to display all design windows with their name. Now let's return to the Tree view.
3. Choose **View** from the GUI Designer menu bar and **Tree** followed by **Icons and Text**.

The Project View is refreshed and plus signs (+) are displayed next to the window parts. A plus sign in a tree view indicates there are more items, or in the case of the GUI Designer, parts in the window.

4. To display the additional parts, click on the plus sign with the left mouse button to **expand** it. Another icon with name **CANxxxxx** is shown. This is the **Canvas** part of the window.
5. Click on the "+" sign beside the **CANxxxxx** icon. The Canvas part is expanded to show all of the parts you have created so far on this window.

Tip

You can also change the Project View by clicking with the right mouse button on the Project View. A pop-up menu is displayed from which you can choose a particular view.

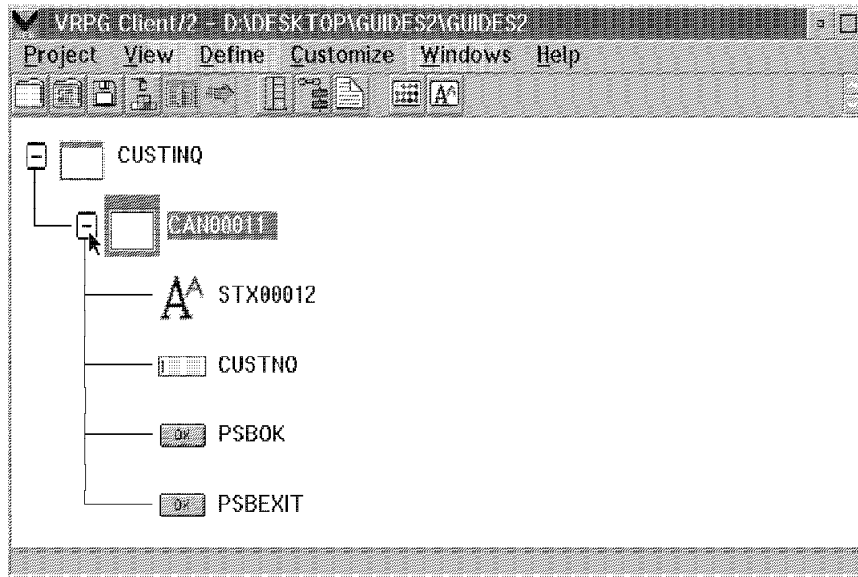


Figure 39. Tree View of the Project

Customizing the Tool Bar

The Tool Bar can be customized by adding and removing icons. To see how you can customize the **Tool Bar**, you add a new icon to it. Perform the following steps:

1. Choose **Options** from the GUI Designer menu bar and **Tool Bar** followed by **Change....** The **VRPG Client - Customize Tool Bar** dialog is shown.

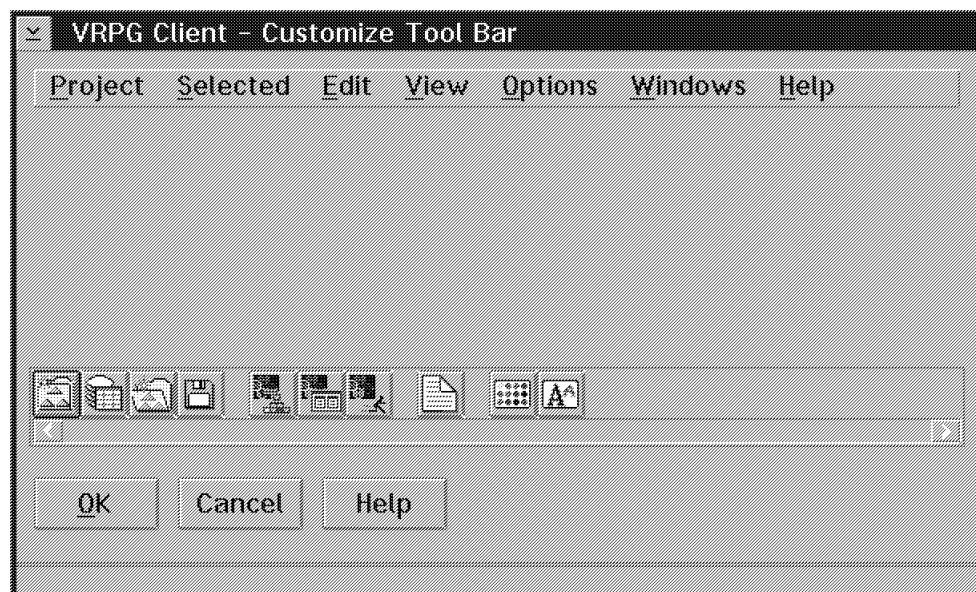


Figure 40. Customizing the Tool Bar

This dialog shows the current state of the Tool Bar on the GUI Designer along with a copy of the GUI Designer menu bar.

2. Choose the **Project** menu bar choice and the **Define reference fields...** menu item. A new icon is added at the end of the Sample Tool Bar.

3. Move the mouse pointer over this new icon and press and hold the right mouse button.
4. Drag it to the left of the **Color Palette** icon and release the right mouse button.

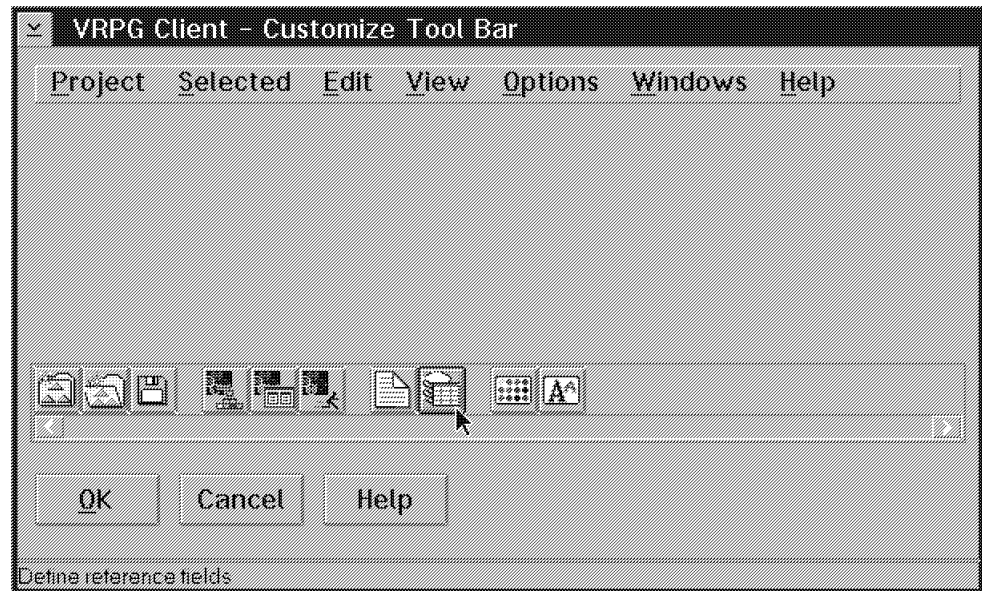


Figure 41. Moving a Tool Bar Push Button

The position of the new **Define reference fields** icon is changed accordingly.

5. Move the mouse pointer to this icon and invoke its pop-up menu by clicking on the right mouse button.
6. Choose the **Add separator** menu choice. A space is added between the **Reference field list** icon and the **Color Palette** icon.
7. Press the **OK** push button to apply the changes to the Tool Bar.
8. Notice that the GUI Designer menu bar has been updated to reflect your changes.

Customizing the Parts Palette

You can customize the **Parts Palette** to contain only those parts that you use most often.

1. Find the **Group box** part on the **Parts Palette**.
2. Invoke its pop-up menu by clicking on the right mouse button while the mouse pointer is over it.
3. Choose **Remove**.

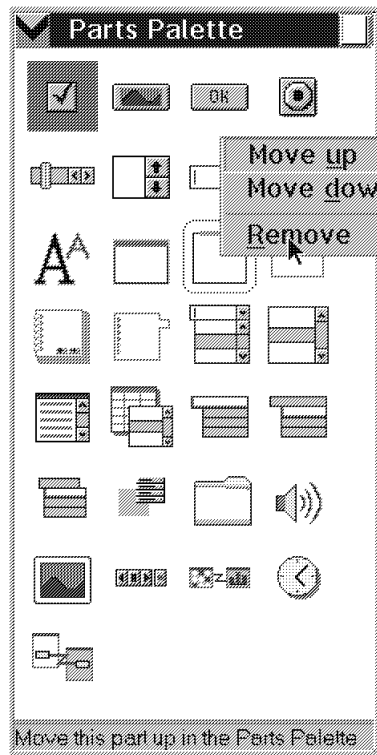


Figure 42. Customizing the Parts Palette

The **Group box** part has been removed from the **Parts Palette**. If you exit the **VRPG Client GUI Designer** at this point, the **Parts Palette** does not contain the **Group box** part when the **GUI Designer** is invoked again.

However, the **Group box** part has just been removed from the Parts Palette, not the GUI Designer. To add the Group box part to the Parts Palette, follow these steps:

1. Click on the icon in the top right corner of the **Parts Palette**.
2. The Parts Catalog is displayed.
3. Select the **Frames** tab. As you can see, the **Group box** part item is still present in the **Parts Catalog**.

Note: As the **Group box** part is an IBM-supplied part, you are not allowed to modify it (for example, change its name or icon) or to delete it from the **Parts Catalog**. You can perform those actions on **User-Defined Part** and **Imported** parts only.

4. Invoke the pop-up menu for the **Group box** part.
5. Choose **Add to palette**.
6. Click on the icon in the top right corner of the **Parts Catalog**.
7. The Parts Palette is displayed with the Group box.

Creating a User-Defined Part

When you are designing windows for your application, you are using parts supplied with VRPG Client. In addition to these parts, you can create your own parts and add them to the **Parts Catalog** and **Parts Palette**. These are called **User-Defined Parts**.

You are going to enlarge the Customer Inquiry window to accommodate some new parts:

1. Double-click on the **CUSTINQ** icon on the **Project View** to open the **Customer Inquiry** design window if it is not currently open.
2. Move the mouse pointer over the upper border of the window. It changes to a double-arrow.
3. Press and hold the right mouse button and size the window to double its vertical size.

This is to size the window so additional parts can be added. In this case, we are adding a logo for a fictitious company. The logo consists of an image part and a static text part.

1. Find the **Image** part on the **Parts Palette** and position the mouse pointer over it.
2. Press and hold the right mouse button and drag-and-drop the **Image** part above the **Enter Customer Number:** Static Text in the **Customer Inquiry** window.
3. Release the right mouse button.
4. Invoke the Settings Notebook for this part by double-clicking on it.

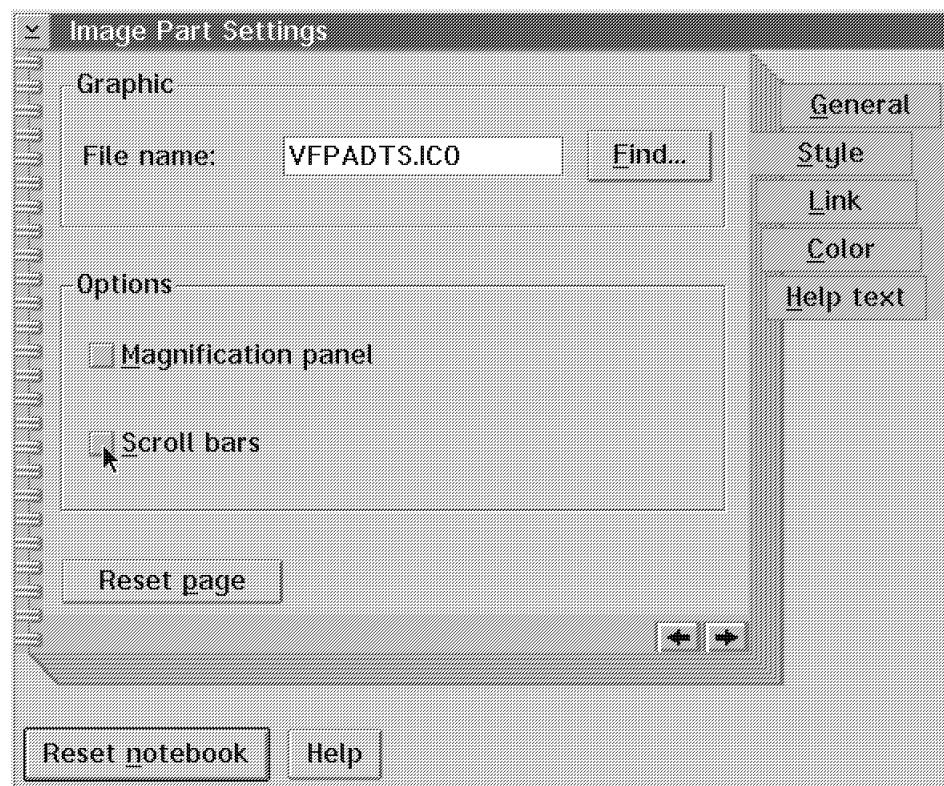


Figure 43. Setting Image Part File Name

5. Go to the **Style** page of the Settings Notebook by selecting the **Style** tab.
6. Type **x:\CODE\EVFPADTS.ICO** as the **File name**. This is an icon that is shipped in the VRPG Client directory. Replace the **x** with the correct drive letter.

Tip

To determine the drive and directory where VRPG Client has been installed, type the SET VRPGDIR command in an OS/2 window.

7. Deselect the **Magnification panel** as well as the **Scroll bars** setting by clicking on the appropriate check boxes.
8. Double-click on the **System** icon of the Image part Settings Notebook to close it.
9. Move the mouse pointer to the lower right sizing handle of the **Image** part. The mouse pointer changes to a double-arrow.
10. Press and hold the right mouse button and size the **Image** part so that the icon fits into it.

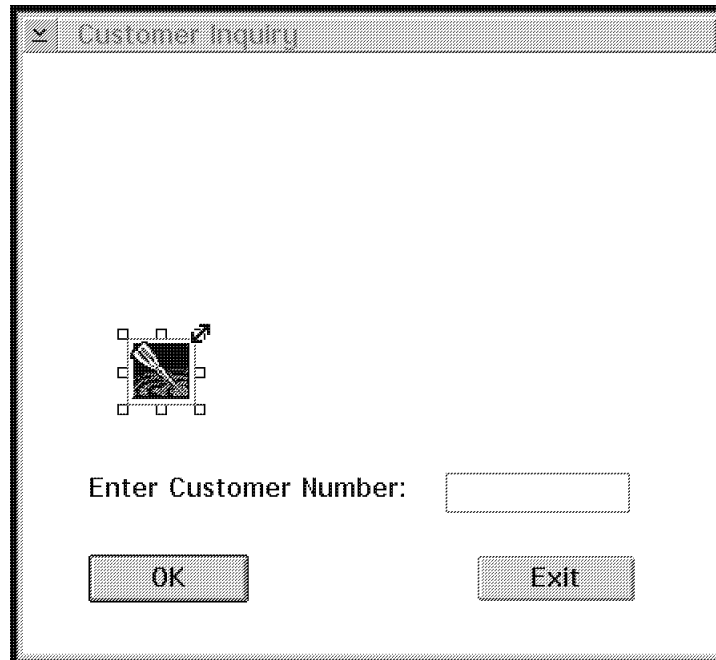


Figure 44. Sizing the Image Part

To add the company name:

1. Find the **Static Text** part on the **Parts Palette** and position the mouse pointer on it.
2. Press and hold the right mouse button and drag-and-drop the **Static Text** icon next to the **Image** part in the **Customer Inquiry** window.
3. Release the right mouse button.
4. Invoke the **Settings Notebook** for this part by double-clicking on it.
5. Change the **Text** setting to **Client Server Applications Inc.** (or any name you want to give your fictitious company).
6. Go to the **Font** page of the Settings Notebook.

7. Select the **Select font** radio button. The **Change font...** push button is enabled.
8. Press the **Change font** push button. The **VRPG Client - Change Fonts** window is shown.
9. Click on the button on the right side of the **Name** drop-down combination box. A list of available fonts is shown.
10. Scroll through the list and select the **Times New Roman** font by clicking on it.
11. Change the font size to **14** in the same way.

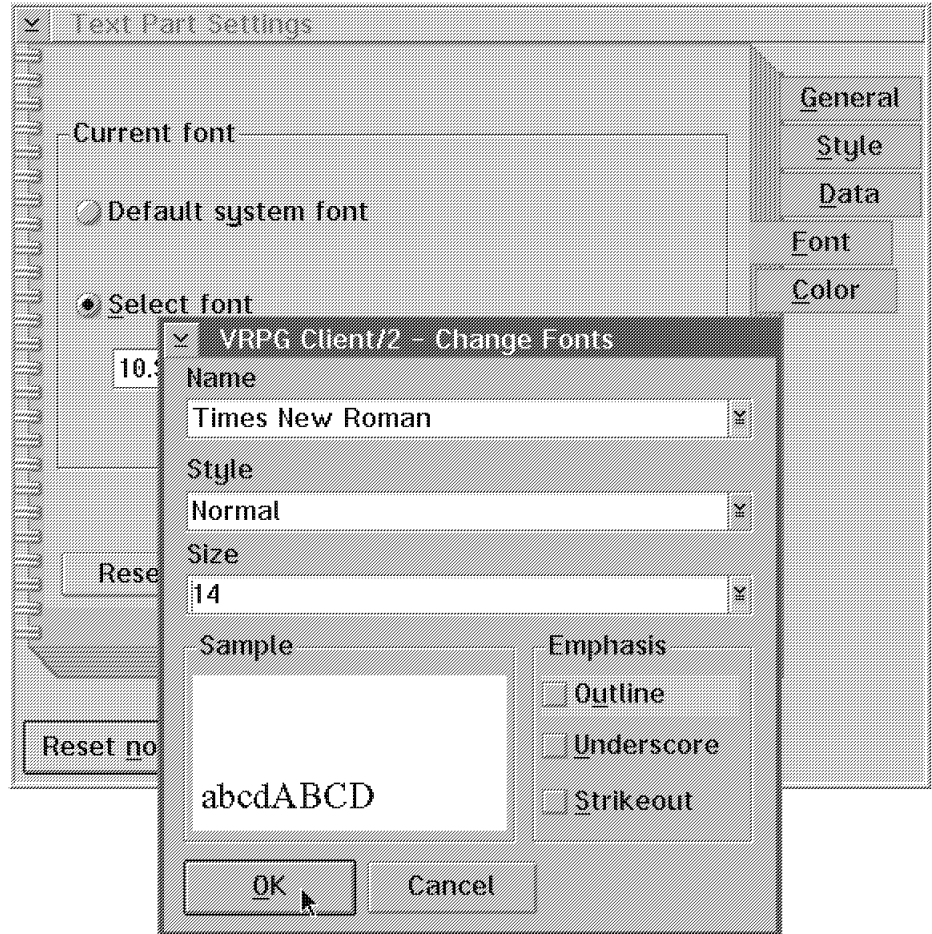


Figure 45. Change Fonts Setting

12. Press the **OK** push button.
13. To change the parts color, go to the **Color** page of the Settings Notebook.
14. Select the **Apply to Foreground** radio button.
15. Select the **Predefined colors** radio button. The combination box containing all available colors is enabled.
16. Select **Red** from the combination box.
17. Go to the **Style** page.
18. From the **Vertical alignment** options, select **Center**.
19. Double-click on the **System** icon of the Static Text part Settings Notebook to close it.

Because the company name does not fit in the default size of the **Static Text** part, you need to enlarge this part:

1. Click on the **Static Text** part holding the company name to make it the active part. The borders of the part are marked by placing the **Sizing Handles** around it.
2. Move the mouse pointer to the lower right sizing handle so that the mouse pointer becomes a double-arrow.
3. Press and hold the right mouse button and move the mouse pointer to the right and down until the entire company name is visible.
4. Release the right mouse button.

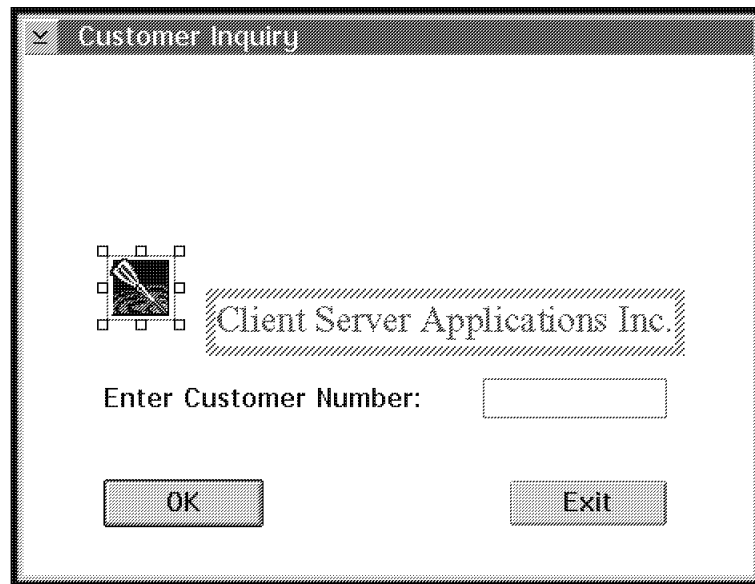


Figure 46. Before aligning the new parts

Now the image part and the company name should be aligned with each other. Using the mouse is the easiest way to select them, but in this exercise, you use the keyboard instead. It is important to know this method since there may be times when using the mouse selects parts you do not want to be aligned. For example, there may be a part between two parts you want to align without affecting the intervening part.

1. Select the **Image** part with the mouse pointer.
2. Press the right arrow key until the **Static Text** part with the company name is highlighted.
3. Press and hold the CTRL key and press the Space bar. This selects the **Static Text** part without de-selecting the **Image** part.
4. Get the pop-up menu of the **Image** part by pressing the right mouse button while the mouse pointer is placed over the **Image** part.
5. Choose **Align** and the second alignment choice offered to horizontally align the company name to the company logo.
6. Re-size the window if necessary so that all of the contained parts reside in the middle of the window.

The intent of this exercise is to use these two parts as a logo that are also reused on other windows. To do this, you create a User-Defined Part from these parts and add it to the Parts Palette and Parts Catalog. This allows you to add your companies logo and name by just dragging-and-dropping the new **User-Defined Part** onto any design window.

To create the User-Defined Part:

1. Move the mouse pointer to the left and above the **Image** part.
2. Press and hold the left mouse button and select the **Image** part containing the company's logo and the **Static Text** part containing the company's name by dragging the mouse so that both parts are within the Selection Rectangle.
3. Release the left mouse button.
4. Get the pop-up menu of either of the two parts.

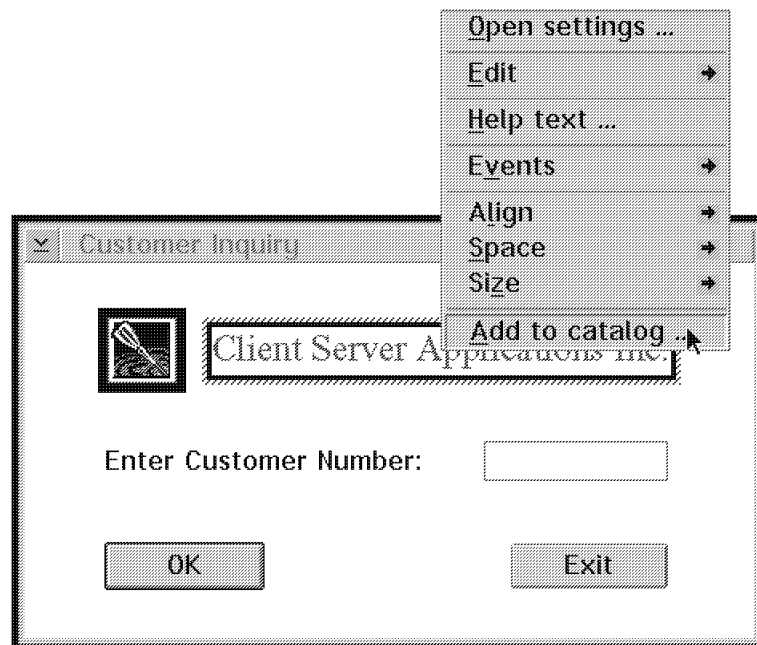


Figure 47. Adding a User-Defined Part to the Parts Catalog

5. Choose **Add to catalog**. The **Create User-Defined Part** window is shown.
6. Change the **Part name** to **Company**.
7. Press the **Find...** push button to locate an icon to represent this User-Defined Part. This icon is shown on the **Parts Catalog** and **Parts Palette**. The **Find an icon for the part** dialog is shown.
8. Type **x:\CODE\EVFPADTS.ICO** in the **Filename** entry field and press the Enter key. Replace the **x** with a valid drive letter. If the drive letter is not known, ask the instructor. You return to the **Create User Defined Part** window. The selected icon is displayed.

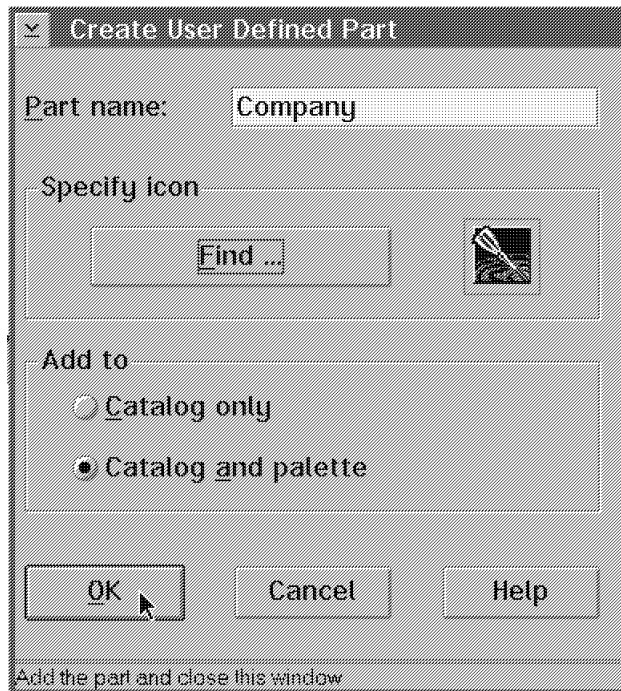


Figure 48. Create User-Defined Part Window

9. Press the **OK** push button to add the new part to the **Parts Catalog** and the **Parts Palette**.
10. Size the **Parts Palette** so that you can see the new **User-Defined Part**.

Note: As you move the mouse pointer over the User-Defined Part on the Parts Palette, its name appears on the Parts Palette information area.

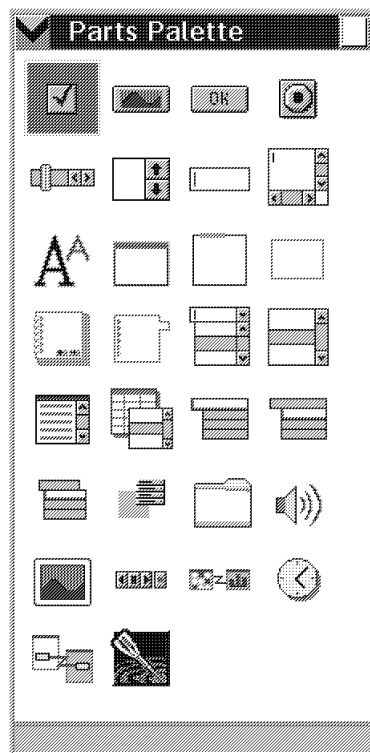


Figure 49. Parts Palette with a User-Defined Part

Close the **Customer Inquiry** window:

1. Invoke its pop-up menu by moving the mouse pointer to the **Title Bar** and pressing the right mouse button.
2. Choose **Close**.

Adding a New Window to Your Project

During this exercise, you create a new **Window with Canvas**, change its attributes, and add parts to it. This window shows the customer information for the customer whose number was typed in the **CUSTINQ** window:

To create the new window:

1. Find the **Window with Canvas** part on the **Parts Palette** and position the mouse pointer on it.
2. Press and hold the right mouse button.
3. Move the mouse until the **Window** part is placed somewhere within the **Project View** of the **VRPG Client** window.
4. Release the right mouse button. A new design window is shown and an icon is placed in the **Project View**.
5. Move the window so that it fits on the desktop if necessary.

To change the settings for the new window:

1. Double-click on the window title bar to open its **Settings Notebook**.
2. Change the **Part name** setting on the **General** page of the Settings Notebook to **CUSTINFO** and its **Title** setting to **Customer Information**.
3. Select the **Start up** tab.
4. Deselect the **Open immediately** option of the **Display style** settings. This allows your program to control when the window is displayed.

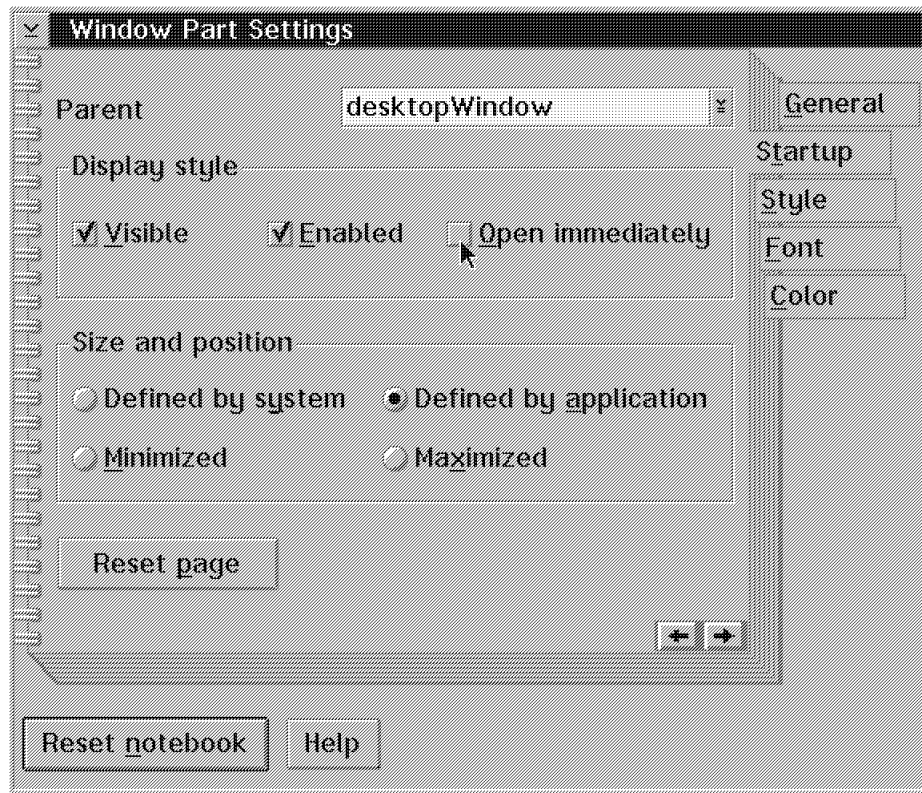


Figure 50. Window Settings Notebook

5. Double-click on the **System** icon of the Window part Settings Notebook to close the notebook.

Creating an Instance of a User-Defined Part

Now you add the company logo and name to this window as you did for the **Customer Inquiry** window at the beginning of this exercise. This time, however, you use the **User-Defined Part** you previously created.

1. Find the **Company** User-Defined Part on the **Parts Palette**.
2. Drag-and-drop this part on the **Customer Information** window below the title bar.

Using an AS/400 File as a Field Reference File

During this exercise, you learn how to add fields to the GUI using an AS/400 file as a **Field Reference File** using the **Define Reference Field** function of VRPG Client. You are using a file on the AS/400 system that contains the necessary fields required for the **Customer Information** window.

The following entry fields are added to this window:

- Customer number
- Customer name
- Contact person
- Phone number
- Customer address consisting of address, ZIP code, and city
- ZIP location

Creating Field Headings

First, you create the field headings for these entry fields by using the Static Text part:

1. Find the **Static Text** part in the **Parts Palette**.
2. Drag-and-drop it at the left side of the window below the **Company** part.
3. Press and hold the ALT key and select the **Static Text** part with the left mouse button. The text becomes editable.
4. Type **Customer Number** as the text of the **Static Text** part.
5. Apply this text to the **Static Text** part by clicking the left mouse button anywhere outside of the part.
6. Select the **Static Text** part and re-size it with the right mouse button so the entire text is displayed.

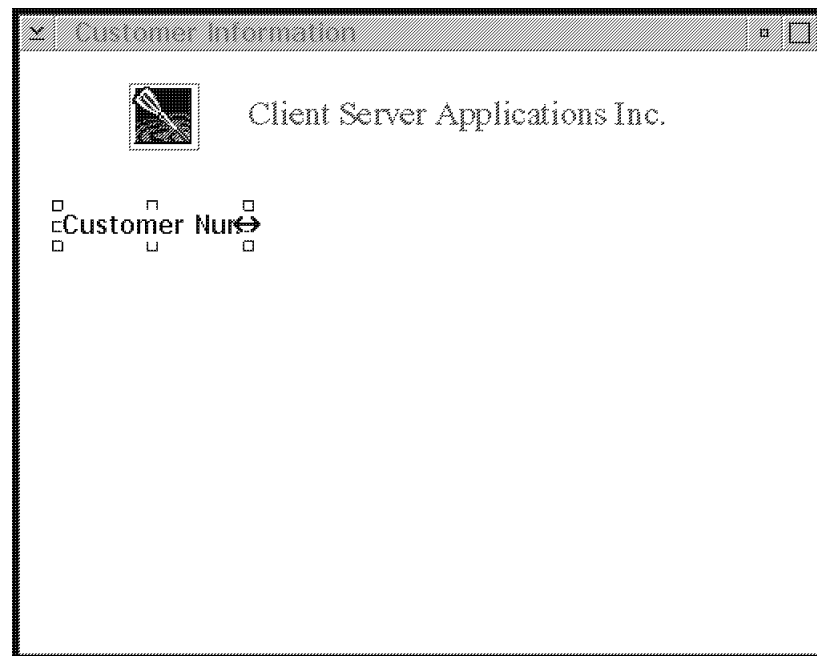


Figure 51. Re-Sizing Static Text Part

Repeat these steps for the other fields of the **Customer Information** window. Do not worry about the alignment of the parts at this point. It is done later. After adding the **Static Text** parts, the **Customer Information** window should look similar to the following example window.

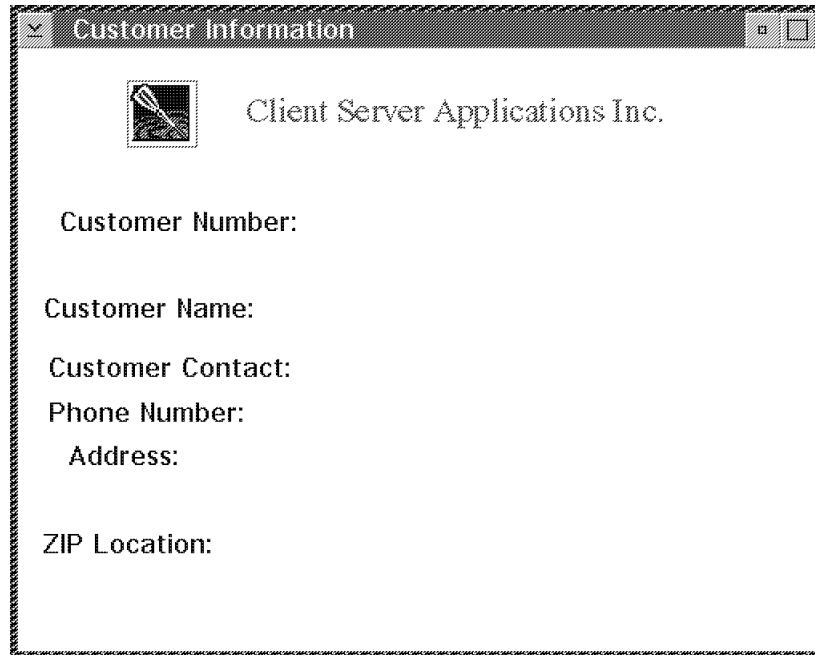


Figure 52. Design Window with Logo and Field Headings

Adding the Entry Fields

You now add the entry fields to the Customer Information window. This time, however, rather than using the Parts Palette, you create the entry fields using the **Define Reference Fields** feature of VRPG Client. This feature allows you to create entry fields by directly referencing fields in a DB2/400 database file.

Defining the Server

Before you can use the **Define Reference Fields** feature, you have to tell the GUI Designer which server or AS/400 system you want to use:

1. Choose **Project** from the GUI Designer menu bar.
2. Choose **Define servers....**
3. Type in the **Server name** as given to you by your instructor. The input area is on top of the list box.

Tip

The value typed here must be a valid PLU name as defined in Communications Manager. You can determine an active 5250 session PLU name by observing the name in the middle of the status line of a 5250 session window. The session must be at the Sign-on display to see the PLU name.

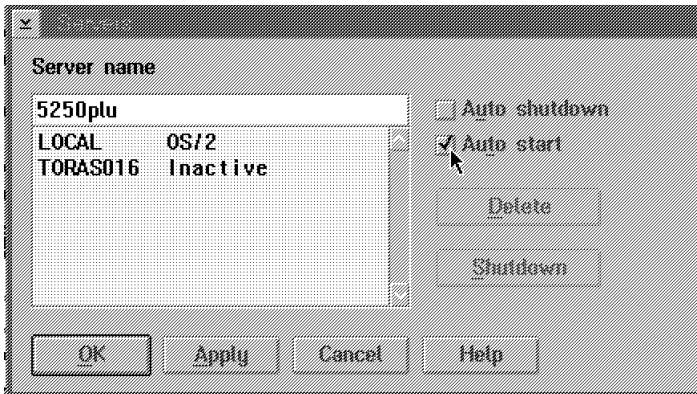


Figure 53. Specifying a Server Name

4. Select the **Auto start** check box.
5. Press the **OK** push button.

For the field reference file, a file named **CUSTOML3** is used. It is located in library **GuiDes2** on the AS/400 server you just specified. To obtain a list of the fields in this file:

1. Choose the **Project** menu item from the GUI Designer menu bar.
2. Choose the **Define reference fields...** menu item. The **VRPG Client - Define Reference Fields** window is shown.

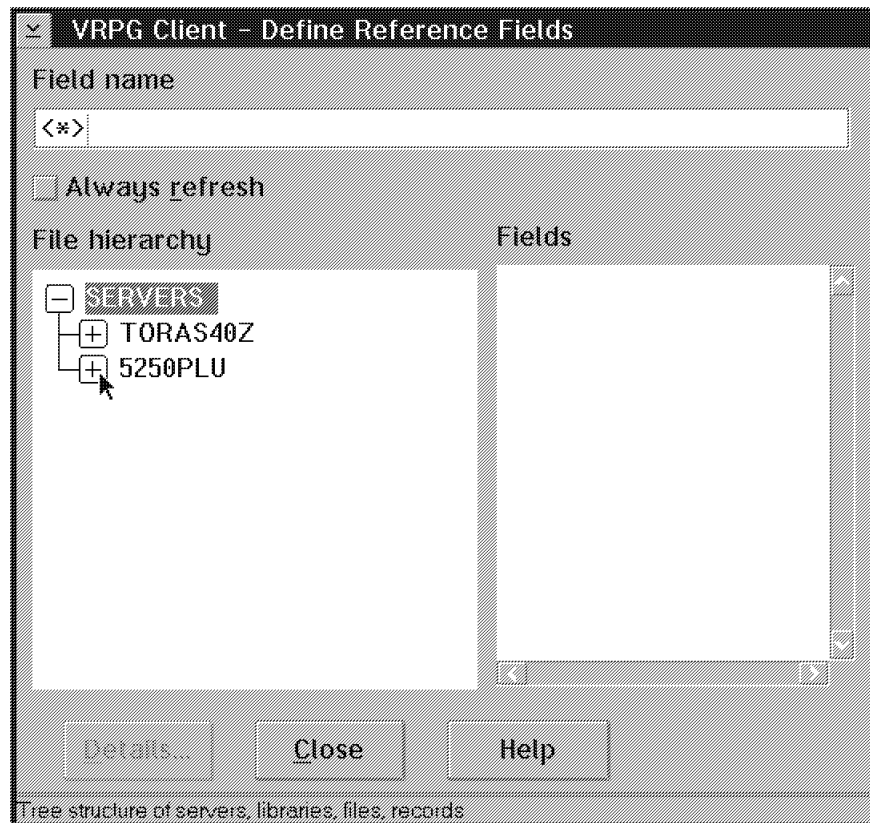


Figure 54. Server List

Note: This list is different on your workstation. Select the server you specified in an earlier step.

- Expand the correct server name in your server tree by clicking on the plus (+) sign next to its name in the **File hierarchy** list. Since this is the first time you are accessing this server, the **VRPG Client - Logon** dialog is shown asking you for your **AS/400 User ID** and **Password**.
- Type **A8LABxx** for both of them where xx is the number assigned to your team by the instructor.

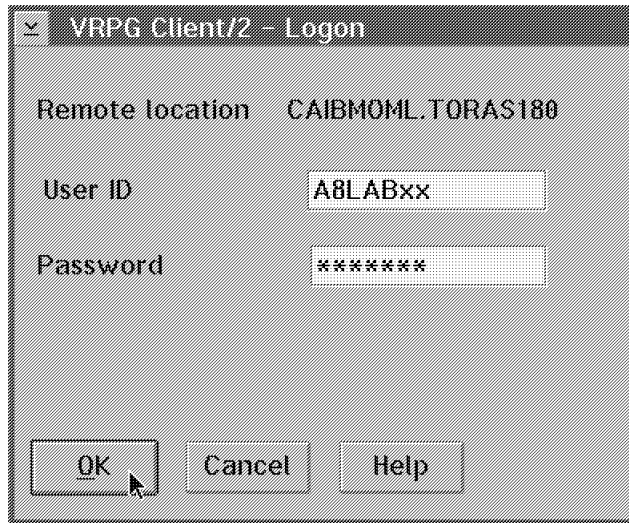


Figure 55. Logon Window

- Press the **OK** push button to log onto the AS/400 system. If the logon is successful, the libraries of your initial library list are displayed (*LIBL).
- Find the library **GuiDes2** in this list and expand it by clicking on the "+" sign. The logical and physical files in this library are displayed.

Note: If the library **GuiDes2** does not appear in the list of libraries, it may not be in your library list. In that case, type **GuiDes2** in the entry field at the top of this window next to the server name and press Enter.

- Click on the "+" next to **CUSTOML3**. A list of the record formats for this file is shown.
- Double-click on the only available format name **CUSTOM01**. All fields in this record format are shown in the **Fields** list.

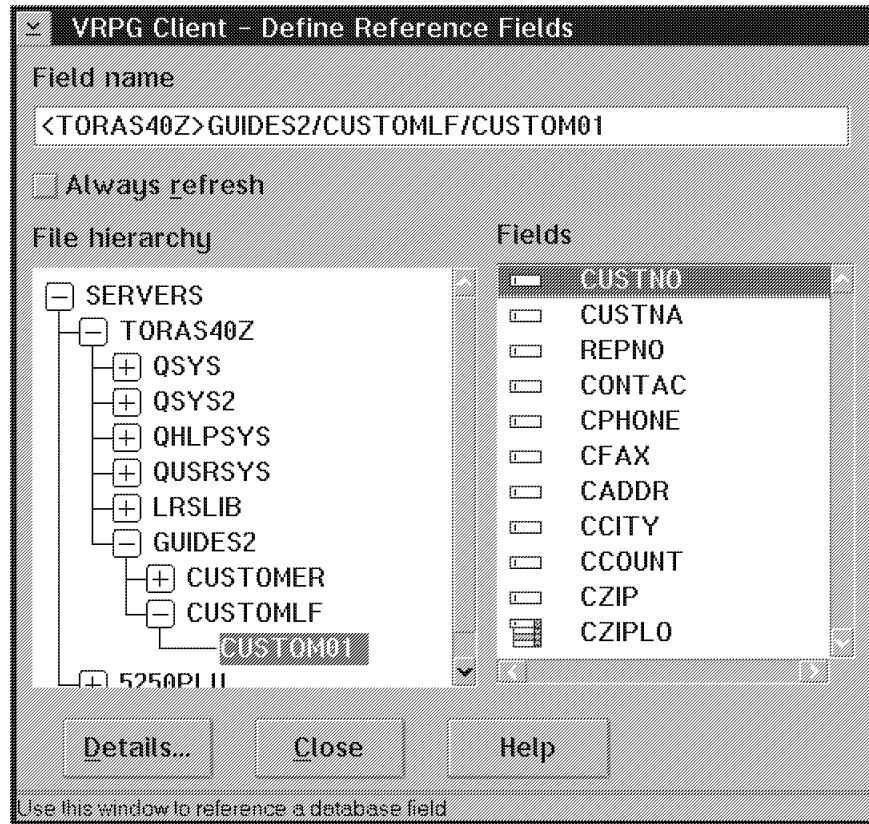


Figure 56. Define Reference Fields - Field List

9. Double-click on the first field **CUSTNO**. The **VRPG Client - Database Field Description** window is shown with detailed information about the field.
10. Press the **OK** push button to close this window.

You can now create the necessary **Entry Field** parts for the **Customer Information** window using some of the fields of the **CUSTOM01** record format as references:

1. Rearrange the **Customer Information** window and the **VRPG Client - Define Reference Field** window so that you can see them both.
2. Move the mouse pointer over the **CUSTNO** entry in the **Fields** list.
3. Press and hold the right mouse button and drag the mouse pointer next to the **Customer Number** Static Text part on the **Customer Information** window.

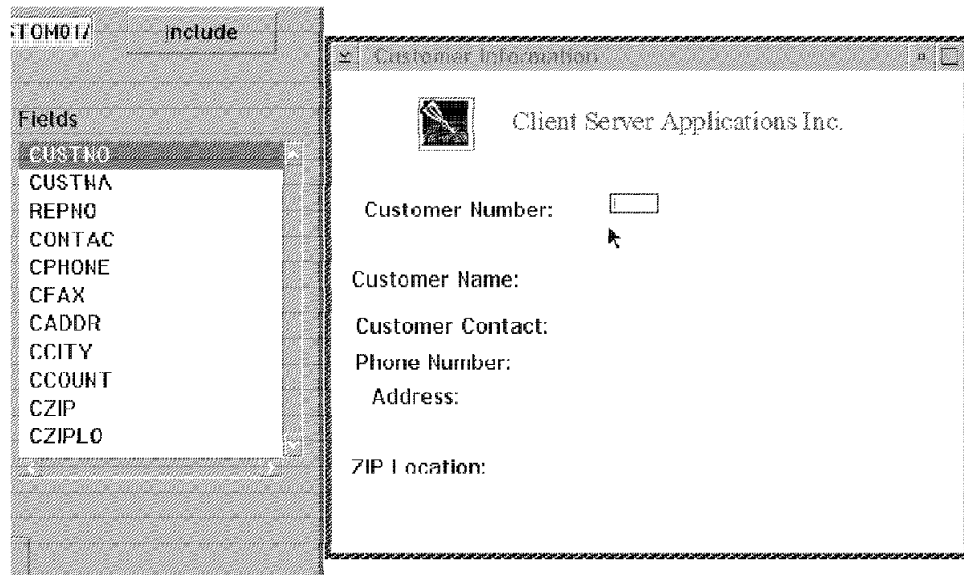


Figure 57. Dragging a Reference Field

4. Release the right mouse button. An **Entry Field** part is created.
5. Double-click on this new **Entry Field** part to invoke its **Settings Notebook**.
6. Go to the **Reference** page of the Settings Notebook. This page shows from which AS/400 database file this field is referenced.

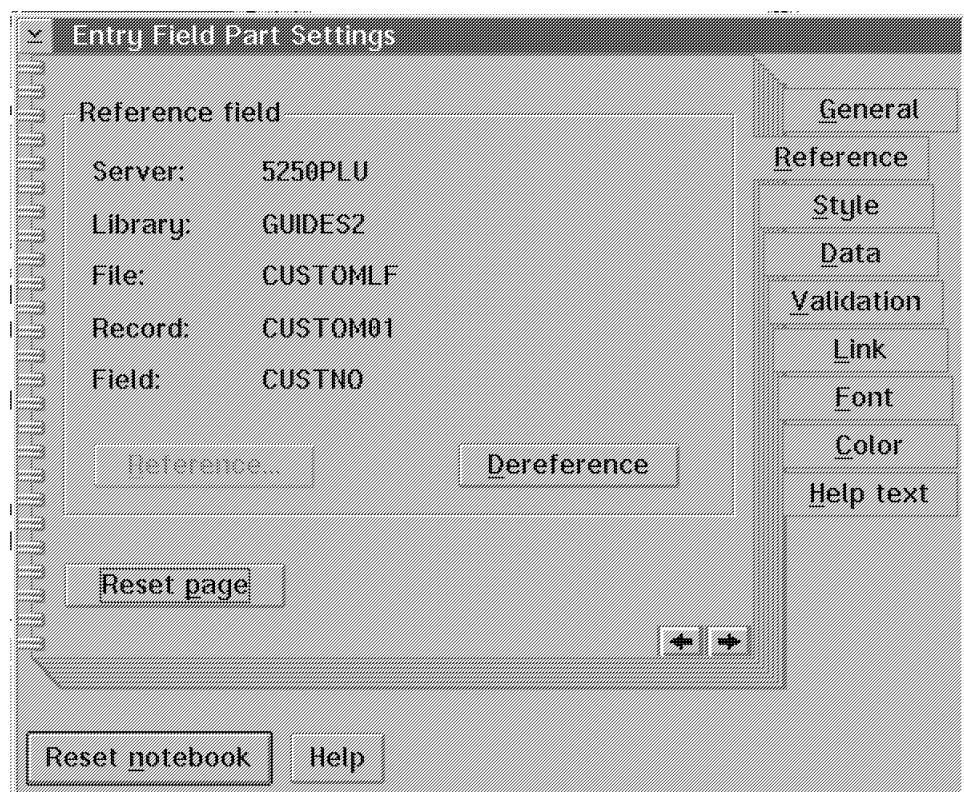


Figure 58. Reference Field Page

7. Go to the **Style** page of the Settings Notebook and select the **Read only** check box to prevent the customer number from being changed.

8. Go to the **Data** page of the Settings Notebook and verify that these attributes have been adopted from the referenced field.
9. Double-click on the **System** icon of the **Settings Notebook** to close it.

Create the other **Entry Field** parts for the **Customer Information** window in the same way using the following AS/400 fields as a reference:

- CUSTNA
- CONTAC
- CPHONE
- CADDR
- CZIP
- CCITY
- CZIPLO

Tip

Several fields can be created from the Define Reference Field window at one time by selecting each field that is required and using drag-and-drop. The drag-and-drop icon indicates several fields are being created.

An example of this window is shown in the following figure.

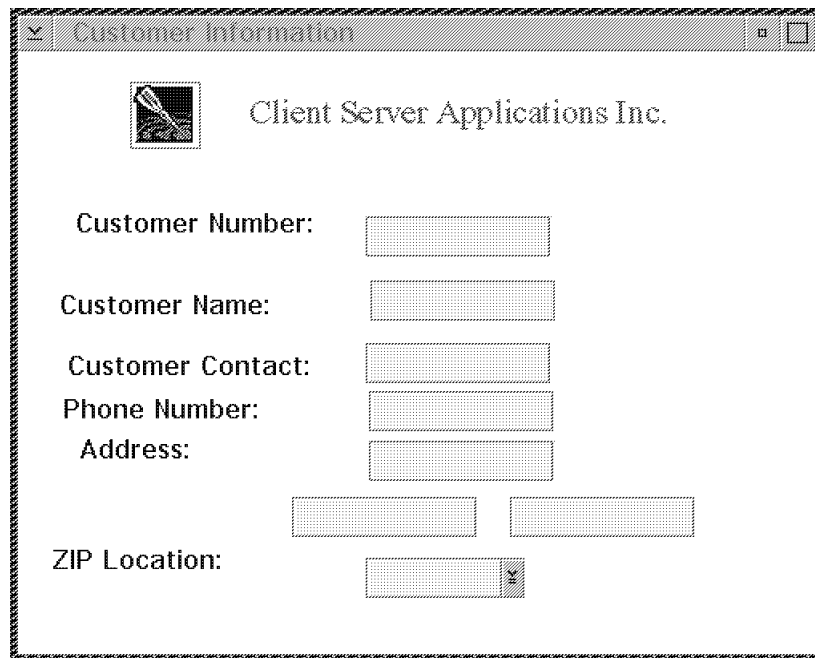


Figure 59. Window with Entry Fields

Note that the part created from the ZIP location field **CZIPLO** is not an **Entry Field** part, but a **Combination box** part. This is because the **VALUES** DDS keyword is specified for this field. You can check this by looking at the details for this field (double-click on **CZIPLO** in the **Fields** list of the **VRPG Client - Database Reference Field** window).

Close the **VRPG Client - Database Reference Field** window by pressing the **Close** push button.

More Alignment, Sizing, and Spacing

Note: Feel free to skip some of the alignment and spacing exercises suggested here if you feel comfortable with using these tools.

Now that all the parts required to make up the **Customer Information** window are created, they still need to be arranged within the window. First, you align all **Static Text** parts containing the description of the information shown in the **Customer Information** window to the same vertical axis:

1. Move the mouse pointer above the **Static Text** part **Customer Number** and to the left of all of the **Static Text** parts.
2. Press and hold the left mouse button and drag to create a Selection Rectangle that it covers all **Static Text** parts.
3. Release the left mouse button. All **Static Text** parts become highlighted.
4. Get the pop-up menu of the **Customer Number** Static Text part by clicking on the right mouse button with the mouse pointer located somewhere over this part.
5. Choose **Align** and the fourth alignment choice offered. All **Static Text** parts are aligned to the left.

Tip

In case you selected the wrong alignment option and the selected parts are overlaying each other, you can use the **Undo** function in the GUI Designer to restore the parts to their original position. To undo the alignment, choose **Edit** from the menu bar and choose **Undo**.

The next step is used to even out the spaces between the Static Text parts **Customer Name**, **Customer Contact**, **Phone Number**, and **Address**.

1. Select these **Static Text** parts in the same way you just did for vertical alignment.
2. Invoke the pop-up menu of one of the parts.
3. Choose **Space** and the fourth menu item.

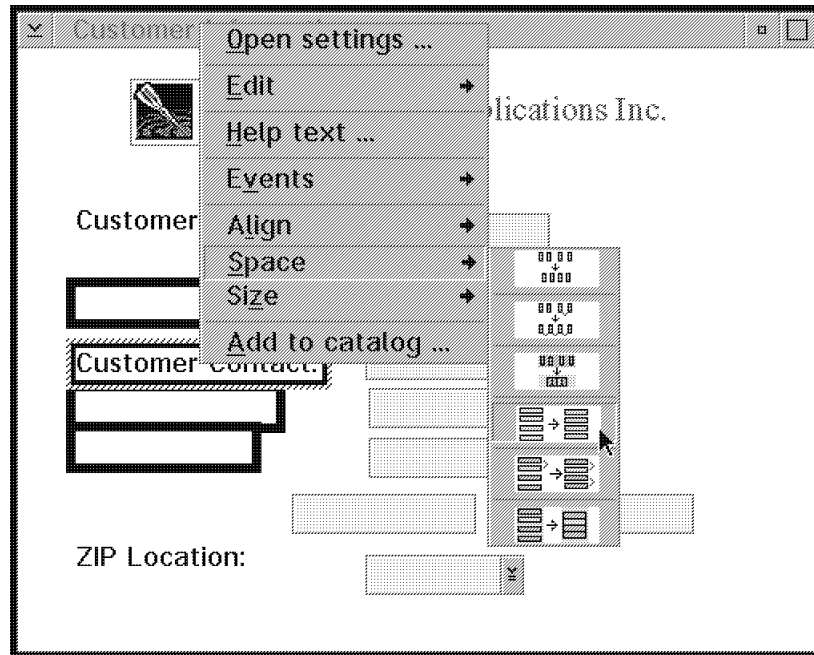


Figure 60. Spacing Static Text Parts

This evens out the spaces between the selected parts.

Now you align each **Entry Field** part with its corresponding **Static Text** part:

1. Draw a Selection Rectangle around the **Customer Number** Static Text part and the **CUSTNO** Entry Field part.
2. Invoke the pop-up menu for the **Customer Number** Static Text part.
3. Choose **Align** and the second alignment choice. The selected parts are aligned to an imaginary horizontal line running through the center of the **Customer Number** Static Text part.
4. Repeat these steps for the **Customer Name** Static Text part and the **CUSTNA** Entry Field part as well as for the **Address** Static Text part and the **CADDR** Entry Field.
5. For the **ZIP Location** Static Text part and the **CZIPLO** combination box part, you have to select the first alignment choice offered to get them aligned properly. This is because a **Combination box** part consists not only of an Entry Field, but also of the value list, which makes it vertically larger than a **Static Text** part.

Note: You might need to drag the Selection Rectangle outside the **Customer Information** window to get it selected.

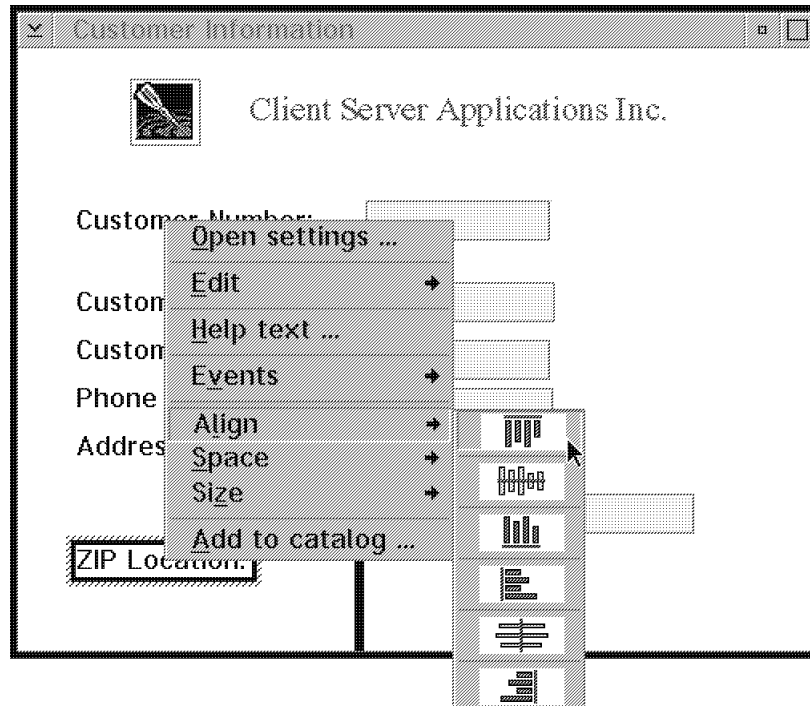


Figure 61. Alignment

To align the **CONTAC** Entry Field part as well as the **CPHONE** Entry Field part to their corresponding **Static Text** parts, you can use the **Spacing** tool again.

1. Draw a Selection Rectangle around the **Entry Field** parts **CUSTNA**, **CONTAC**, **CPHONE**, and **CADDR** parts.
2. Get the pop-up menu of any of the selected parts, choose **Space**, and the fourth choice offered. The spaces between the **Entry Field** parts are evened out.

Now align all **Entry Field** parts except **CCITY** to one vertical axis:

1. Draw a Selection Rectangle around all of the Entry Field parts with the exception of the **CCITY** part.

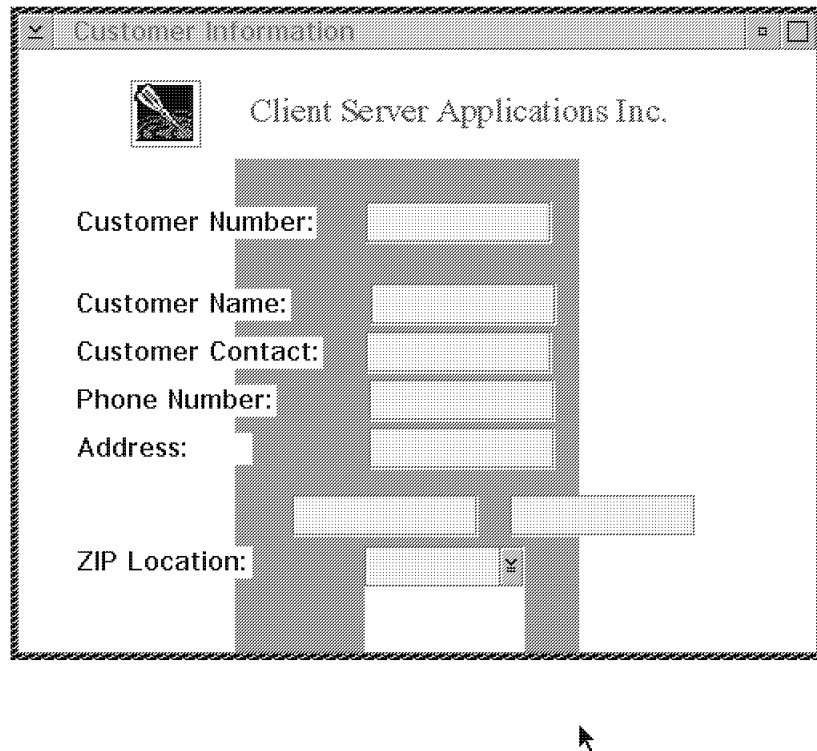


Figure 62. Left Align Entry Fields

2. Choose the fourth choice out of the **Align** menu item from the pop-up menu of the **CUSTNO** Entry Field part.
3. If the **CCITY** Entry Field part became overlapped by the **CZIP** Entry Field part, move the **CCITY** Entry Field part to the right of the **CZIP** Entry Field part.

To change the horizontal spaces between the **Entry Field** part **CZIP** and the **Combination box** part **CZIPLO** on one hand and the **Entry Field** part **CADDR** on the other hand to be equal to the space between the **Entry Field** parts **CPHONE** and **CADDR**:

1. Draw a Selection Rectangle around the **Entry Field** parts **CPHONE**, **CADDR**, **CZIP**, and **CZIPLO** parts.
2. Get the pop-up of any of the selected parts and choose **Space** and the fifth choice offered. This uses the space between the **Entry Field** parts **CPHONE** and **CADDR** and applies it to the other selected parts so they are evenly spaced.

Finally, the **Entry Field** part **CCITY** and the **Static Text** part **ZIP Location** need to be horizontally aligned to the **Entry Field** parts **CZIP** and **CZIPLO**.

1. Draw a Selection Rectangle around the **Entry Field** parts **CZIP** and **CCITY**.
2. Invoke the pop-up menu of the **CZIP** Entry Field part.
3. Choose **Align** and the second alignment choice offered.
4. Repeat these steps for the **ZIP Location** Static Text part to be align to the **CZIPLO** Entry Field part.

The following window is shown:

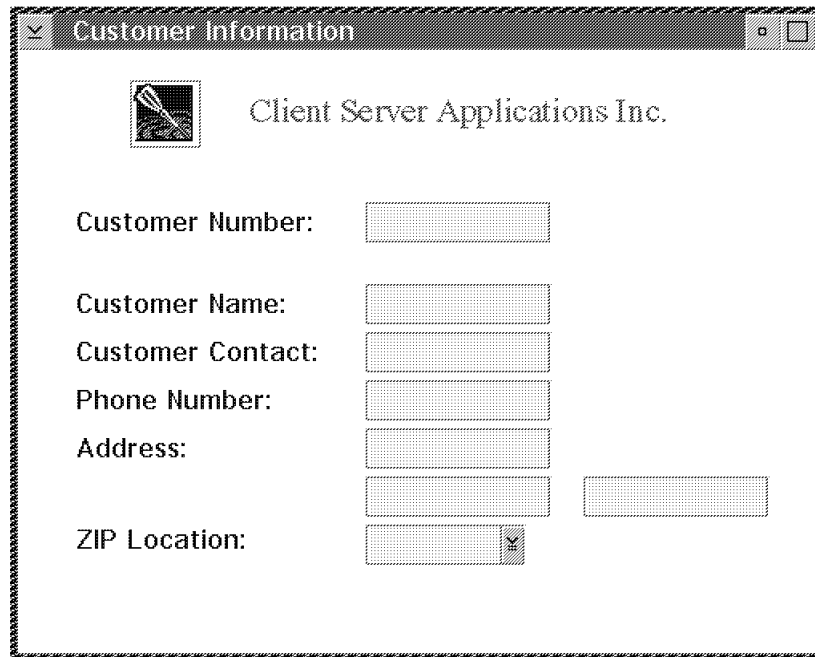


Figure 63. After Alignment

Changing the Size of Some Fields

You might also want to change the size of some **Entry Field** parts that are supposed to display strings but are too long to be displayed using the default size of the **Entry Field** part on the window. You do this now for the **Entry Field** parts **CUSTNA**, **CONTAC**, and **CADDR**.

Even though you use the Define Field Reference feature that sizes the fields according to their definition, you might want to go through this exercise to become familiar with the **sizing tools**.

1. Click on the **CUSTNA** Entry Field part to make it the active part.
2. Drag the middle of the right sizing handles until the field size is enlarged to the right border of the **CCITY** Entry Field part.

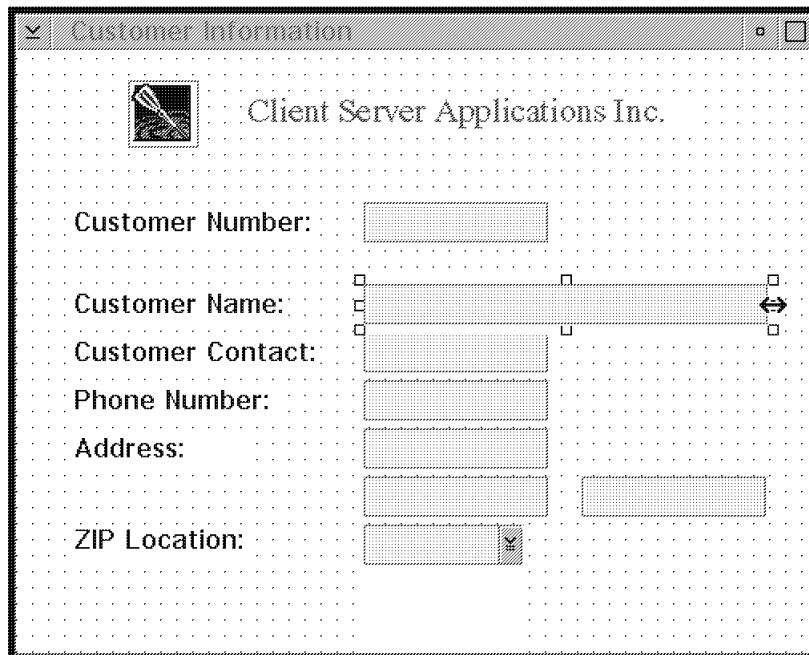


Figure 64. Sizing CUSTNA Entry Field

3. Use the arrow keys to get to the **CONTAC** Entry Field part.
4. Press and hold the CTRL key and press the space bar to select it.
5. Use the arrow keys to get to the **CADDR** Entry Field part.
6. Select this part also by using the CTRL key and the space bar.
7. Invoke the pop-up menu for the **CUSTNA** Entry Field part.
8. Choose **Size** and the first choice offered. The size of the **Entry Field** parts **CONTAC** and **CADDR** are enlarged to the size of the **CUSTNA** Entry Field part.

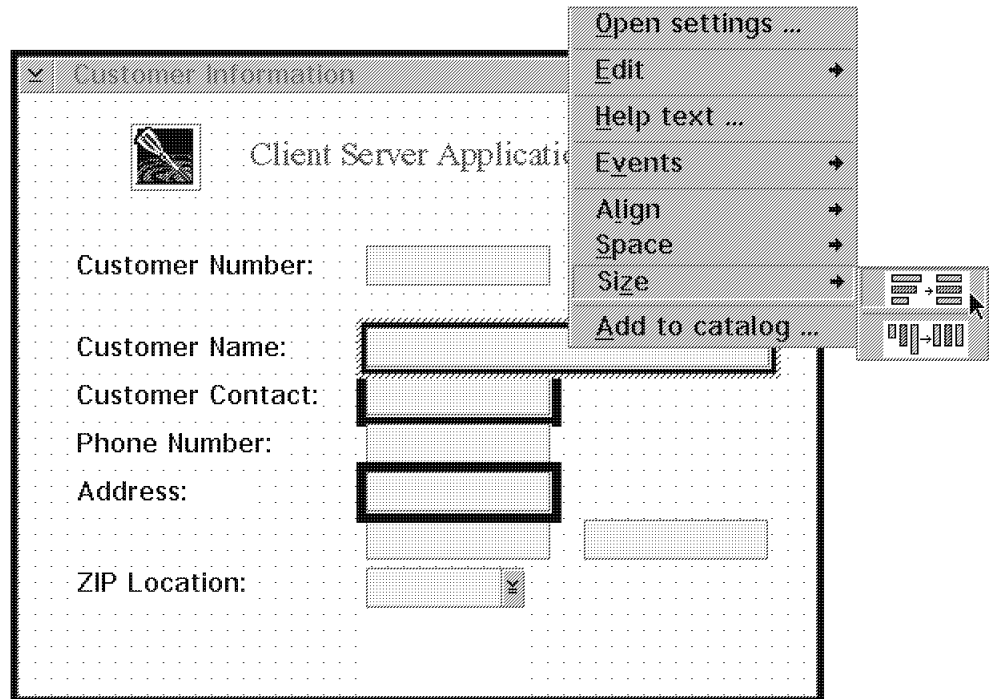


Figure 65. Re-Size CONTAC and CADDR

9. Deselect the **Grid** menu item out of the **Customer Information** window's pop-up menu.

Now you add a **Push button** part to the **Customer Information** window that allows the user to close this window and return to the **Customer Inquiry** window.

1. Move the mouse pointer to the bottom border of the **Customer Information** window so that it becomes a double-arrow.
2. Press and hold the right mouse button and drag the window border so that it can hold the additional **Push button** part.



Figure 66. Re-Size Window

All parts contained in the **Customer Information** window have also been moved to the bottom of the window. This happens because the focal point for all parts in a **Window** part is the lower left corner of the window.

3. Draw a Selection Rectangle around **all** of the parts on the **Customer Information** window.
4. Move the mouse pointer over the **Static Text** part that holds the Company name **Client Server Applications Inc.**
5. Press and hold the right mouse button and drag the **Company name** Static Text part to just below the **Title bar** of the **Customer Information** window.
6. Release the right mouse button. All of the selected parts have been moved.
7. Find the **Push button** part on the **Parts Palette**.
8. Drag the **Push button** part below the **ZIP Location** Static Text part and release the right mouse button.
9. To align the push button with the Static Text part above, select the **ZIP Location** Static Text part and the **Push button**.
10. Invoke the pop-up menu of the **Static Text** part.
11. Choose **Align** and the fourth alignment choice offered. The **Push button** part gets vertically aligned to the **Static Text** parts of the window.
12. Double-click on the **Push button** part to get the **Settings Notebook** for this part.
13. Change the **Part name** setting to **INFOOK** and its **Text** setting to **OK**.
14. Go to the **Style** page of the Settings Notebook.
15. Select the **Default** check box. This causes the push button's **Press** event Action Subroutine to be invoked when the **Enter** key is pressed.

16. Double-click on the **System** icon of the **Settings Notebook** to close it.

The window should look similar to this:

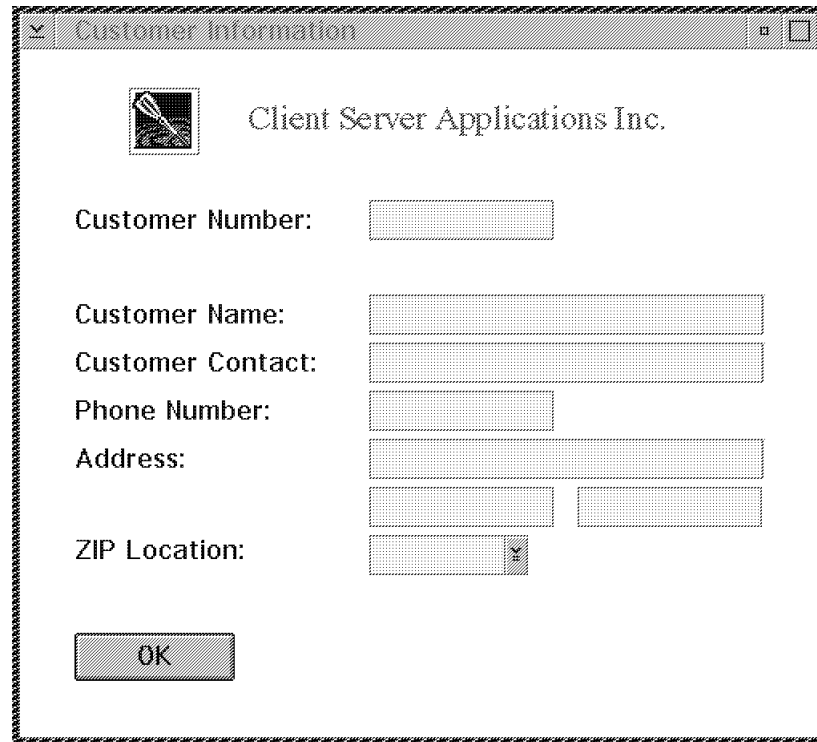


Figure 67. Final Window Design

To save the changes to your project and exit the **GUI Designer**:

1. Press the **Save Project** icon on the **Tool Bar** (the third icon if you have not customized it) and your project is saved.

Adding More Logic to Your Project

The objective of this exercise is to add the necessary logic to use the Customer Information window you have just designed with customer data.

There are several steps to follow:

1. Add a file definition to your program.
2. Change code for the OK push button Press event on the CUSTINQ window to display the CUSTINFO window.
3. Create code for the OK push button Press event on the CUSTINFO window to close it.
4. Build the application.
5. Run the application.

The first step is to define the DB2/400 file in your program:

1. Choose **Project** option from the GUI Designer menu bar and choose **Edit source code**.
2. Scroll to the **H** specification.

3. Position the mouse cursor on the H specification.
 4. Choose **Edit** from the LPEX Editor menu bar and choose **Insert Prompt**.
 5. Press the **Select** push button and select the **F**, specification radio button. Press the **OK** push button and you see the prompt window for an F specification.
 6. Fill in the following values:
 - File name: CUSTOML3
 - File type: I
 - File designation: F
 - File Format: E
 - Record address type: K
 - Device: Disk
 - Keywords: Remote
- Note:** With the exception of the **REMOTE** keyword, all entries are the same as RPGIV on the AS/400 system.
7. To save this input, choose **File** from the LPEX Editor menu bar and choose **Save**.
 8. Close the LPEX Editor by double-clicking with the left mouse button on its system menu.

Changing the OK Button Action Subroutine

You now add the logic for the **Press** event of the **OK** push button on the **CUSTINQ** design window.

Note: Before you go ahead and code the **real** logic for this Action Subroutine, you may want to remove the three lines of code from the previous lab, **Lab 1**. That code was only created to demonstrate events and Action Subroutines. If you prefer the push button to change color, just leave this code in.

The RPG logic you are coding now should perform the following functions:

1. Get the customer number from the Entry Field on the CUSTINQ window.
2. Retrieve customer detail from the AS/400 database file.
3. Display the second window (CUSTINFO).
4. Put data from the database file onto the second window.

Let's get started:

1. Using the pop-up menu from the **OK** push button, bring up the Action Subroutine for the **PRESS** event. You should see the code you created in an earlier lab that changed the push buttons color.
2. The first step is to get the customer number entered in the CUSTNO Entry Field.
3. There are two methods for getting this value:
 - a. Using the READ operation code:

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+.
CLON01Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+HiLoEq
C                                READ      'CUSTINQ'
```

Figure 68. Using READ Operation Code

b. Using the GETATR operation code:

```
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C      'CUSTNO'      GETATR      'TEXT'      CUSTNO
```

Figure 69. Using the GETATR Operation Code

Note: Code only one of these methods in your Action Subroutine.

4. With the value of the field CUSTNO, you now have to chain to the database file CUSTOML3.

Add the CHAIN statement with an indicator in the HI value column (record not found).

You might want to use F4 for prompting to get the right column for this indicator, or try out the syntax checker in the LPEX Editor.

Your CHAIN statement should look similar to this:

```
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C      CUSTNO      CHAIN      CUSTOML3      50
```

5. Add the following statement, which is true if the CHAIN was successful:

```
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C      If      *IN50 = *OFF
```

6. Now you have to code a SHOWWIN operation code to get your second window displayed:

```
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C      SHOWWIN      'CUSTINFO'
```

7. You also have to get the data from the RPG program storage to this CUSTINFO window, which is similar to a WRITE to a record format in an AS/400 display file.

So code:

```
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C      WRITE      'CUSTINFO'
```

8. Now code the following to complete the IF condition:

```
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C      ENDIF
```

Your completed code should look similar to the following example:

```

CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C      PSBOK          BEGACT   PRESS      CUSTINQ
C      Read           'CUSTINQ'
C      CustNo         CHAIN     CUSTOML3      50
C      If             *IN50 = *OFF
C      ShowWin        'CustInfo'
C      Write          'CustInfo'
C      EndIf
C      ENDACT
C
*
```

Figure 70. Using the GETATR Operation Code

This completes this Action Subroutine for now. You can add some more function after we look at the run-time behavior. To save your changes, choose **File** from the LPEX Editor menu bar and choose **Save**. Close the LPEX Editor by double-clicking with left mouse button on its system menu.

Adding an Action Subroutine to the CUSTINFO Window

In this exercise, you add some code to make the CUSTINFO window invisible because the window should disappear when the OK button is pressed.

The instructions in the following exercises are not as detailed because you are now familiar with the **VRPG Client GUI Designer** development environment.

Perform the following steps:

1. Create an Action Subroutine for the **PRESS** event of the **OK** push button on window **CUSTINFO**.
2. Add a statement to set the **Visible** attribute for the window **CUSTINFO** to off.

Tip

All of the VRPG Client manuals are online and can be accessed by choosing **Help** from the GUI Designer menu bar.

3. Save this Action Subroutine.

You might want to close the design windows at this point. To close a design window, position the mouse pointer on the windows title bar and click the right mouse button. From the pop-up, choose **Close**.

Building the Application

Before you build the application, you have to specify where the compiler can find the AS/400 information it needs at compile time. In this case, you have to define on which server (AS/400 system) the file your program is using is located and which file you are using.

To enter this information, choose **Project** and **Define AS/400 information...** from the GUI Designer menu bar. The **Define AS/400 Information** notebook is displayed. This notebook has several pages on which you define information regarding your server and files.

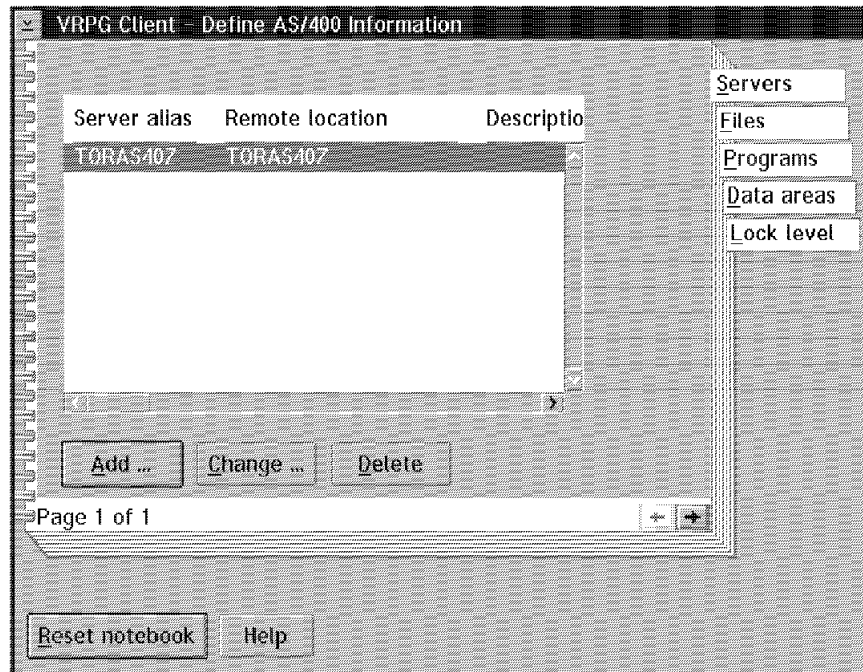


Figure 71. AS/400 Information

For this lab, you need to define a server and a file.

1. On the **Servers** page, press the **Add...** push button. The **Add Server Alias Name** dialog is shown. Type an alias name in the **Server alias** Entry Field. From the drop-down combination box, select the same server you used when using the Define Reference Fields function.
2. Press the **OK** push button to add this server alias to the list on the Server's page.
3. Go to the **Files** page.
4. Press the **Add...** push button to add a new file definition. The **Add File Alias Name** dialog is shown.
5. In the **File Alias** Entry Field, type the name of the file you specified on the **File** specification in your program (CUSTOML3).
6. In the **Remote File** Entry Field, type the actual file name (GUIDES2/CUSTOML3).
7. From the **Server Alias** drop-down combination box, select the server alias name you defined in a previous step.
8. Press the **OK** push button to add this file definition.
9. Close the **Define AS/400 Information** dialog by double-clicking on its system menu.

With this information set, you can now build the project.

1. To build (compile) the project, choose **Project** from the GUI Designer menu bar, choose **Build**, and choose **OS/2**.
2. If you made changes to any of the design windows, a message window is displayed prompting you if the project should be saved first. Answer **Yes** to this message.

3. A status window is displayed indicating that the build is in progress.
4. Once the build is complete, a completion window is displayed indicating if the build was successful or not.

Viewing the Compile Listing

Recall that the VRPG Client compiler is resident on the workstation. Therefore, the compile listing is also resident on the workstation. At any time, it is possible to view the files that make up your project including the compile listing.

To display the compile listing for this latest build:

1. Choose **Windows** on the GUI Designer menu bar.
2. From the pull-down, choose **Workframe project**.

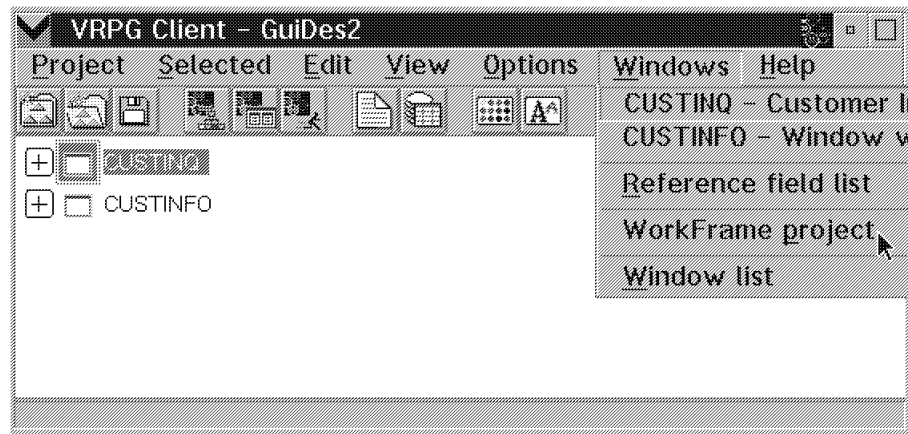


Figure 72. Displaying the Project Files

3. This shows an Icon view of the files that make up a project. Find the one that has a ".LST" file extension and select it with the left mouse button. Click with the right mouse button to get a pop-up menu for the actions allowed for this file. Choose **Edit**.

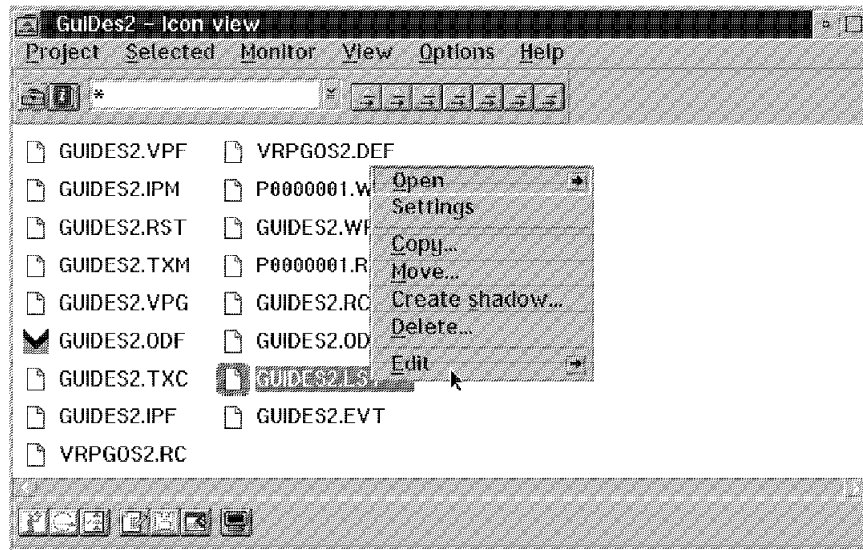


Figure 73. Icon View of the Project

4. The compile listing is now displayed. Note the similarities between the VRPG Client compile listing and the compiler output of the AS/400 RPG compilers.

Testing Your Application

1. If the compile is successful, you can run your program. There are two ways to do this:
 - a. Choose **Project** from the GUI Designer menu bar, choose **Run**, and choose **OS/2**.
 - b. Press the **Run** icon on the Tool Bar (the running man).
2. The first window should be displayed. Type **0010100** as the customer number. Notice that you have to move your cursor to the field before typing in the number. In the next lab, you will improve this.
3. Press the **OK** push button. This brings up the second window with the customer information filled in.
4. If you type in a wrong customer number, nothing happens since you did not code any error handling.
5. Go to the second window again and use the OK button; the window is invisible.
6. Now try to get to the second window again with a valid customer number.
7. You receive a **run-time error** because the **SHOWWIN** operation code is trying to create a window that has already been created.
8. The next lab shows why this is happening and how to handle it.
9. Now press the Exit button to terminate the application.

Lab 3: Enhancing Your Application by adding Error Handling, Action Links, and Messages

In this Lab, you learn to handle error conditions and deal with the run-time behavior of the VRPG Client environment. You also learn about action links and VRPG Client messages. You do this by:

1. Removing the run-time error when using SHOWWIN twice
2. Positioning the cursor on an Entry Field
3. Linking two events to one Action Subroutine
4. Creating messages and displaying message boxes
5. Debugging your application
6. Building the application
7. Running the application

What You Should Be Able to Do

As a result of this Lab, you are able to:

- Handle run-time errors in your application.
- Understand how to position the cursor.
- Link multiple events to one Action Subroutine.

- Use messages and message boxes.
- Use the VRPG Client debugger.

Enhancing the Run-Time Behavior

In this exercise, you are enhancing the application that you built during the previous labs. The run-time error and cursor positioning problems are fixed.

Handling the Run-Time Error

When you ran your application, a run-time error was displayed when you pressed the **OK** button in window **CUSTINQ** the second time. This is because the **SHOWWIN** operation code is executed again that attempts to **Create** the **CUSTINFO** window again. Since it has already been created, the run-time error occurs. The run-time error is displayed because you have not coded any error handling in the application.

As is a typical technique in RPG, you use an error indicator on the operation code to deal with the error, and have a conditional statement to handle the error depending on the state of the error indicator.

The logic inside the **IF** condition uses the **SETATR** operation code to make the **CUSTINFO** window visible if the **SHOWWIN** fails. Remember that pressing the **OK** push button on the **CUSTINFO** window made this window invisible.

Let's update your logic to handle the run-time error:

1. Edit the Press Action Subroutine for the **OK** button in window **CUSTINQ**.
2. Add an **Error Indicator** to the **SHOWWIN** operation code statement. This traps the run-time error. You now have to make the window visible but you do not want to re-create it.
3. Add a conditional statement for checking the error indicator. If the indicator is on, you want to make the window visible and give it focus.
4. Add code to set the **Visible** attribute for window **CUSTINFO** to **1**.
5. Now you need to add a line of code to give the window focus. By giving the window focus, it is the top window. To give a part focus, you set its **FOCUS** attribute to **1**.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
CLON01Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+HiLoEq
*
C      PSBOK      BEGACT   PRESS      CUSTINQ
C              Read      'CustInq'
C      CustNo     Chain    CustomLF      50
C              If        *IN50 = *OFF
C              ShowWin   'CustInfo'      51
C              If        *IN51 = *ON
C              Eval      %Setatr('CustInfo':'CustInfo':'Visible')=1
C              Eval      %Setatr('CustInfo':'CustInfo':'Focus')=1
C              EndIf
C              Write     'CustInfo'
C              EndIf
C              ENDACT
*
```

Figure 74. Error Handling Code

6. Do not forget to add the **ENDIF** operation code to complete the IF condition.
7. Save your changes.

Now you add code to position the cursor in the **CUSTNO** Entry Field. For this exercise, you use the windows **Activate** event to move the cursor to the Entry Field. This event occurs when the user clicks on a window with the mouse.

1. Create an Action Subroutine for the **CUSTINQ** windows **Activate** event.
2. Add a statement that uses the **SETATR** operation code to set the **Focus** attribute of the **CUSTNO** Entry Field to **1**.

```

File Edit View Actions Options Windows Help
Row 12 Column 1 Replace
-----1-----2-----3-----4-----5-----6-----7-----
*
* Change activity:
*
* Who Date Flag Description
*
*
*****
*
C CUSTINQ BEGACT ACTIVATE CUSTINQ
c 'custing' setatr 1 'focus'
c 'custno' setatr 1 'focus'
*
C ENDACT

```

Figure 75. Code to Position Cursor

3. Save your changes and build the project.

Running the Improved Application

1. After successfully building your application, run it again.
2. Enter customer number **0010200** and press **OK** to display the second window with the customer information.
3. Now press the OK button on CUSTINFO; this brings you back to CUSTINQ, but the cursor is **NOT** positioned on the Entry Field.
4. Press the OK button on CUSTINQ again. The run-time error that you got before does not appear so your error handling code is working.

Working with Action Links

We still have the problem of the cursor not going to the **Customer number** Entry Field when the first window is first displayed. You already have the logic to move the cursor in the **Activate** event for the window. Rather than rewrite this logic, you use the **Action Link** feature of VRPG Client. This feature allows more than one event to invoke the same Action Subroutine.

You use this feature to have the **CUSTINQ** windows **CREATE** event call the Action Subroutine already defined for the **ACTIVATE** event.

First let's look at the Action Link window. If an LPEX Editor window is not already opened, open one by choosing **Project** and **Edit source code** from the GUI Designer menu bar.

From the LPEX Editor menu bar, choose **Edit** and **Action subroutines...** The **VRPG Client - Action subroutines : Events View** window is displayed. Enlarge this window so you can see all of the information.

1. From the **Windows** list box, select **CUSTINQ** by clicking on it with the left mouse button.

2. This fills the **Parts** list box with all of the parts on that window.

Since you are interested in the events for the **CUSTINQ** window, select it from the **Parts** list box.

3. You now see the **Events** list box filled with all valid events for a window.

Choose the **Create** event in the Events list box and select **CUSTINQ+ACTIVATE+CUSTINQ** in the **Action subroutines** list box.

The **Create link** push button is now enabled.

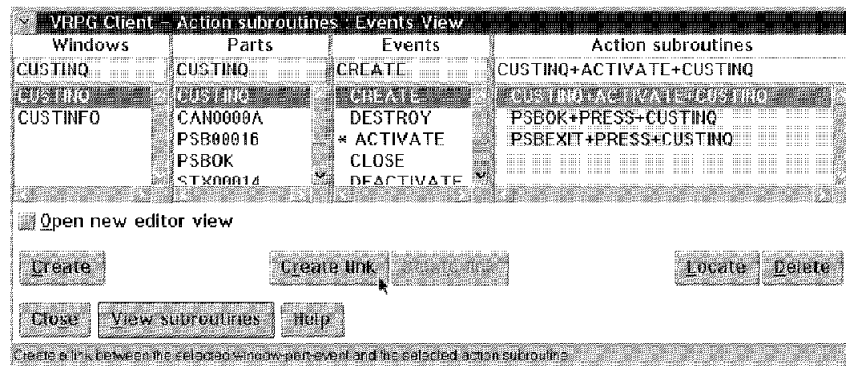


Figure 76. Creating an Action Link

4. Press the **Create link** push button. This link the **Create** event to the **Activate** event Action Subroutine.
5. Close this window by pressing the **Close** push button.
6. Build your application and test it by noting that the cursor is positioned in the CUSTNO field when the window is first displayed.

Using Message Boxes

As the application is now, if the user enters a customer number that is not in the database, no action is performed and no message is displayed. In the next exercise, you add logic that causes a message box to be displayed to inform the user that the customer number is invalid.

There are two methods of defining the message text for messages in VRPG Client:

1. Define the message text and message box style with D specifications in your VRPG Client source.
2. Define a message in the GUI Designer.

In either case, the **DSPLY** operation code is used to display the message in a message box.

In this exercise, you define the message in the GUI Designer and add a DSPLY opcode to your VRPG Client logic to display a message if the customer number is not found in the database.

Let's start by defining the message:

- Choose **Project** from the GUI Designer menu bar and choose **Define messages....**
- In the **VRPG Client - Define Messages** window, press the **Create...** push button. The following window is displayed that allows you to define a new message.

The screenshot shows a dialog box titled "VRPG Client - Edit Message". It contains the following fields and controls:

- Message ID:** MSG0001
- Type:** A dropdown menu showing "information".
- Message:** A text area containing "Customer number is not valid. Please enter a valid customer number."
- Message Help:** A text area containing "Type additional help here."
- Message Window:** A section containing a checked ☒ **Moveable** checkbox.
- Buttons:** A dropdown menu showing "okButton".
- Default:** Three radio buttons labeled "Button 1", "Button 2", and "Button 3". "Button 1" is selected.
- Buttons:** At the bottom, there are three buttons: "Save", "Cancel", and "Help".

Figure 77. Creating a Message in VRPG Client

- Scroll through the different types of messages that are available and select any one. The message type determines which icon is displayed on the message box.
- Type some message text in the **Message** entry box. This is the text that is shown on the message box.
- You may also type some additional message text in the **Message Help** entry box. This text is shown when the **Help** push button is pressed on the message box.
- Select the **Moveable** check box. If this is not checked, the user cannot move the message box.
- Select the push buttons to appear on the message box from the **Buttons** drop-down combination box. For this exercise, choose just the **OK Button**.
- Press the **Save** push button to save your message definition.
- **MSG0001** now is shown in the **Define Messages** window.

Note: VRPG Client assigns message numbers. Message numbers cannot be changed by the user.

- Press the **OK** push button on the VRPG Client - Define Messages window.

Now, you need to add the code to your Action Subroutine to display the message you have just created.

- Open the **CUSTINQ** window if it is not already opened and choose the **PRESS** event for the **OK** push button to edit its Action Subroutine.
- Add the following statements after the **CHAIN** statement to display the message if the record is not found:

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+.
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C          ELSE
C      *MSG0001      DSPLY          REPLY          9 0
```

Figure 78. Adding the DSPLY Operation Code

- Now add a statement to change the background color of the Entry Field to red using the SETATR operation code.

Tip

The part name goes in Factor 1, the attribute value in Factor 2, and the attribute name in the Result field. Do not forget the quotes.

All of the online manuals including the Parts Reference are also available from the LPEX Editor **Help** menu.

To see a list of attributes that are allowed for a specific part type, select the part on the Parts Palette by selecting it with the left mouse button and press **F1**. Help is displayed for the part along with **hyperlinks** to that part's attributes.

```
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C      ??????      SETATR      ?????????      ?????????
```

Figure 79. Logic to Change the Background Color

- Build and run the application. Enter an invalid customer number, such as 9999999. A message box should be displayed with the message text you defined earlier.

Note: Notice that a **Help** push button has been added to the message box. This is because you specified second-level message text. Press the Help push button to see the second-level help you typed when you created the message.

This is an example of how messages are created in VRPG Client and how they can be displayed with the **DSPLY** operation code. Messages are used elsewhere in VRPG Client including the **Message Subfile** part and label substitution. These areas are not covered in this lab.

Optional Exercise - More Message Box Handling

The first part illustrated how to display a message using the DSPLY operation code and specifying the message number in factor one. There are other ways of displaying a message or a text with the DSPLY operation code.

The next method is similar to the one in the previous exercise. However, instead of specifying a message number in Factor 1, you define the message on a **Definition** specification. The following steps demonstrate how to do this:

- Enter the following specifications in your program:

Notice that the Definition type is **M**. This is unique for VRPG Client, and indicates this definition specification is defining a message.

Tip

D specifications must follow any File (F) specifications and come before any Calculation (C) specifications.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
DName+++++++ETDsFrom+++To/L+++IDc.Keywords+++++++
*
DInfoBox          M          STYLE(*INFO)
D                  BUTTON(*RETRY: *ABORT: *ENTER)
*
DMSG1             M          MSGNBR(*MSG0001)
```

Figure 80. Defining a Message Box

- Replace your current DSPLY operation code with the following code:

```
CLON01Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+HiLoEq
C   MSG1          DSPLY   InfoBox      Reply          9 0
```

Figure 81. Displaying the Message Box

- Build your application and run it. When the message box is displayed, notice that the push buttons and the icon are different than the ones you specified when you created the message. By using this method of displaying a message, you can override the attributes defined when the message was created.

Using Message Substitution

VRPG Client messages also support message substitution. Using message substitution, the value of a program variable can be displayed in the message text.

For this exercise, you create a new message that shows the invalid customer number in the message box.

- Create a new message as you did for the first message. This time, for the message text, use message substitution. Type the following information as the message text:

Customer number %1 is not valid. Please enter a valid customer number.

Note the difference. The message substitution variable **%1** has been added. This variable is replaced at run-time with the value of a variable in your program.

- Add the following D specifications to your program. The keyword **MSGDATA** is used to define which program variable should be used as the message substitution text. In this case, you are using the customer number entered by the user.

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+.
DName+++++ETDsFrom+++To/L+++IDc.Keywords+++++
*
DMSG2          M          MSGNBR(*MSG0002)
D              MSGDATA(CUSTNO)
*
```

Figure 82. Using Message Substitution

- Replace the current DSPLY operation code with the following code:

```
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C      MSG2          DSPLY          Reply          9 0
```

Build your application and run it. Now, when the message box is displayed, the customer number should be displayed in the message box when you enter an invalid number.

Defining Message Text in the Program

An additional variation for displaying a message is to define the text of the message in your VRPG Client program. This is done as follows:

- Add the following D specifications to your program. The **TEXT** field defines the message text to be displayed.

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+.
DName+++++ETDsFrom+++To/L+++IDc.Keywords+++++
*
DWarnBox       M          STYLE(*WARN)
D              BUTTON(*YESBUTTON:*NOBUTTON)
*
DTEXT          M          MSGTEXT('Number NOT found')
*
```

Figure 83. Defining Message Text in the Program

- Replace the current DSPLY operation code with the following code:

```
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C      TEXT          DSPLY      WarnBox      Reply          9 0
```

- Build your application and run it. Observe the message text on the message box.

Using Action Links to Navigate Your Logic

The **Action link** window can also provide you with some help in navigating through your application logic. This window is accessed from the LPEX Editor.

1. If an LPEX Editor window is not currently open, choose **Project** and **Edit source code** from the GUI Designer menu bar. The LPEX Editor window is shown.
2. From the LPEX Editor menu bar, choose **Edit** and **Action subroutines....** The **VRPG Client - Action subroutines : Events View** window is shown.
3. Press the **View subroutines** push button. This gives you a different view of all the action links in your program. Click on the **CUSTINQ+ACTIVATE+CUSTINQ** Action Subroutine and the second list box shows all of the action links for this Action Subroutine. In your case, this should be 2: the **Create** and the **Activate**.

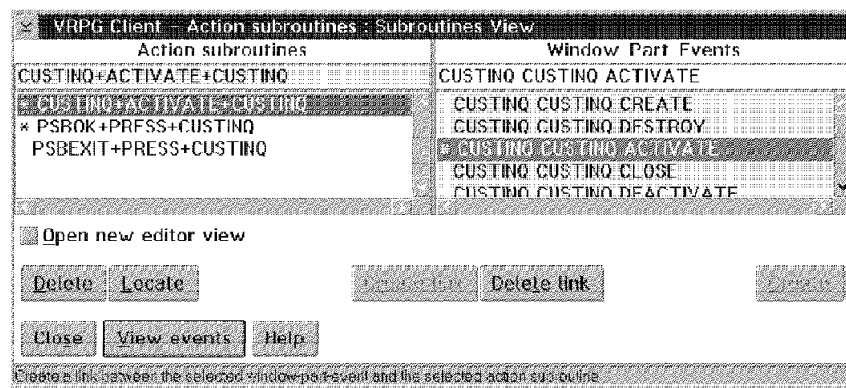


Figure 84. Navigate with Action Link Window

4. To view the logic for an Action Subroutine, double-click on an Action Subroutine name. The LPEX Editor positions the source to that Action Subroutine.

Debugging a VRPG Client Application

VRPG Client includes a powerful debugger to help in locating logic errors in your program. This short exercise illustrates some of the basic debugger features.

The debugger can be started by clicking on the debug **Tool Bar** icon or from the GUI Designer menu bar by choosing **Project**, **Debug**, and **OS/2**.

1. The debugger **Debug Session Control** dialog is displayed followed by another window with the source view of your program.
2. In the source view, scroll to the **Activate** Action Subroutine.

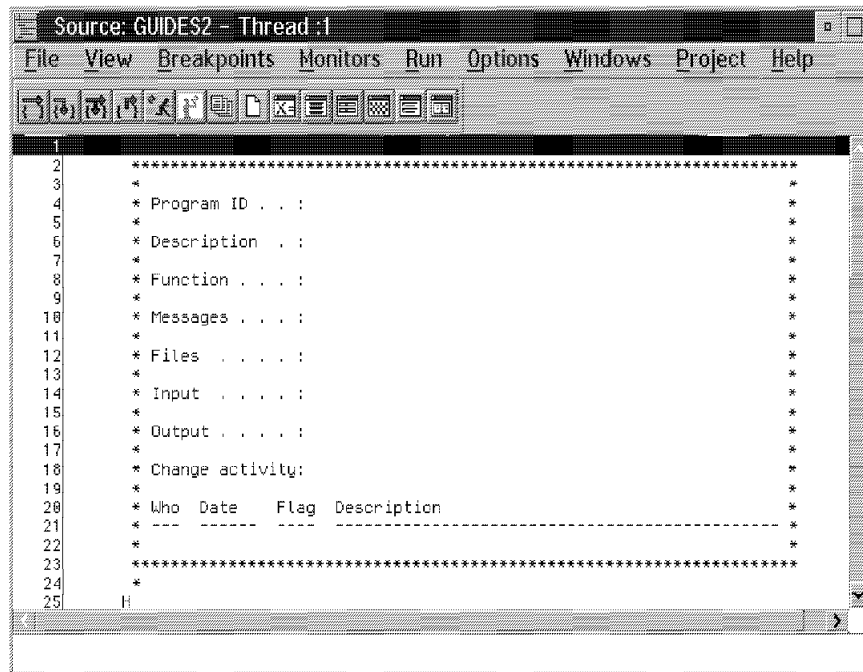


Figure 85. Debug Source View

3. Set a break point on the statement that sets the focus attribute by double-clicking on the statement number.
4. Run your application by pressing the **Run** icon on the source view Tool Bar. The **Run** icon is the one with the green circle and running person.

Tip

As you move the mouse pointer over the Debugger Tool Bar icons, a short description of each icon is displayed.

5. When the break point is reached, the statement is highlighted.
6. To examine the contents of a program variable, scroll to any statement that contains a variable.

Note: This technique does not work on the **EVAL** operation code.

7. Double-click on the variable name. The **Program Monitor** window is shown with the variable contents.
8. Allow the program to run by pressing the **Run** icon on the Debugger Tool Bar.
9. Press the **Exit** button on the CUSTINQ window to end the debug session and your program.

This concludes this lab. In the next, lab you will make more enhancements to your application by adding a **Subfile** part and some **Application Help**.

End the GUI Designer by double-clicking on its system menu.

Lab 4: Creating a component with Subfile and Application Help

In this lab, you enhance an existing application by creating a new component with a Subfile part. In addition, you add some help to your application.

You are also making changes to your current application. Currently, the CUSTINQ window has two push buttons:

DETAIL

EXIT

You add a LIST push button that starts the new subfile component that you are about to create in this Lab.

During this lab you:

1. Create a new component in a project.
2. Create a subfile and add fields.
3. Write the VRPG Client logic to write records to the subfile part.
4. Write the VRPG Client logic to get data from a subfile part.
5. Add application help to your application.
6. Build the application.
7. Run the application.

What You Should Be Able to Do

As a result of this lab, you can write applications containing multiple components, subfile parts, and application help by:

- Adding a new component to an existing project.
- Creating subfiles with multiple fields.
- Writing VRPG Client logic to display data in a subfile part.
- Writing VRPG Client logic to respond to Subfile events.
- Writing application help using the **Information Presentation Facility (IPF)** tag language.

Creating a Subfile

In this exercise, you are enhancing an application that was created in an earlier lab. As the application is now, the user must know ahead of time what a valid customer number is.

The enhancement you make is to add a push button to the first window, **CUSTINQ**, that starts a new component to show another window. That window contains a subfile part that lists all of the customers in the database. When the user selects a customer from the list, the customer number is placed in the Customer Number Entry Field on the **CUSTINQ** window.

In this Lab, you only create the subfile; later in the class, you will have an opportunity to use a Component Reference Part to fill the Customer Number Entry Field on the **CUSTINQ** window.

Note: The instructions in this lab are not as detailed as in the earlier labs since you are expected to have had some experience using the VRPG Client GUI

Designer. Remember that online help is always available from the **Help** menu item on the GUI Designer.

First, you create a new component called **COMPLIST**. These steps guide you through the process of creating a new component in an existing application.

1. Double-click on the Customer Inquiry project on your Desktop.
2. You see all of the parts belonging to the Customer Inquiry in the project window.

If you are a VRPG release 1 user, this looks unfamiliar to you. VRPG Client is now using Workframe V3 and some of the windows have changed.

The project window is the development platform for a VRPG Client project. From here, you can edit, run and debug your application.

3. Click on the **PROJECT** menu bar choice and select **CREATE** from the menu pull-down. If you clicked on the small arrow, select **VRPG new** from there.

This brings up the GUI Designer. You now have to tell it that you want to create a component and also identify the component name. Select the **SAVE PROJECT** icon on the toolbar. You see a window that allows you to specify the component name **COMPLIST**.

4. Type the name **COMPLIST**.
5. Select the Radio button **New Component**.
6. Select the **Customer Inquiry** project from the **Folders and Projects** Combo box.
7. Make sure the source file has COMPLIST as an entry.
8. Click the Ok push button.

You have now set up the component for the subfile window and can go ahead with creating a subfile.

Now you create a subfile that is used to display records from a logical file on the AS/400 system. This file is named **CUSTOML1** and is in library **GUIDES2**. The record format name is **CUSTOM01**. The subfile you create shows data from the following database fields:

CUSTNO
CUSTNA

Perform the following steps to create the new window with the subfile part:

1. Change the title of the design window to **Select Customer** (or something equivalent).
2. Name the design window **CUSTLIST**.
3. Locate the **Subfile** part on the Parts Palette and drag-and-drop it on the design window. Position and size the subfile as required.
4. Double-click on the subfile part to bring up its Settings Notebook.
5. On the **General** page, change the name of the subfile by changing the value in the **Part Name** Entry Field to **SFL1**.
6. Close the Settings Notebook.
7. Now you add fields to the subfile. There are two ways to add fields to a subfile part:

- a. By opening the Subfile parts Settings Notebook and going to the **Field list** page.
- b. By dragging-and-dropping fields from a database Reference File.

For this exercise, we use the database reference method.

Defining the Server

Before you can use the **Define Reference Fields** feature, you have to tell the GUI Designer which server or AS/400 system you want to use:

1. Choose **Project** from the GUI Designer menu bar.
2. Choose **Define servers....**
3. Type in the **Server name** as given to you by your instructor. The input area is on top of the list box.

Tip

The value typed here must be a valid PLU name as defined in Communications Manager. You can determine an active 5250 session PLU name by observing the name in the middle of the status line of a 5250 session window. The session must be at the Sign-on display to see the PLU name.

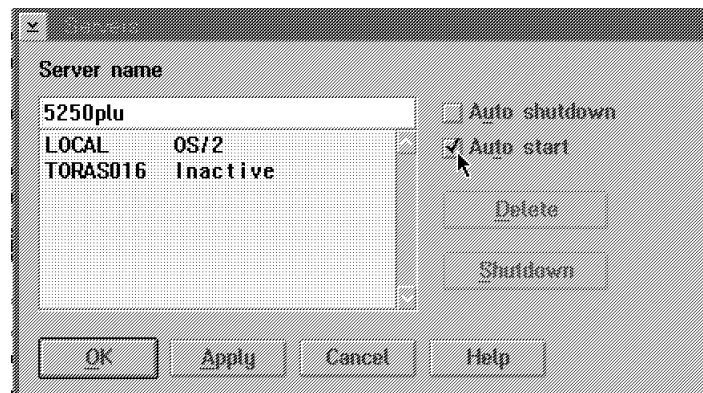


Figure 86. Specifying a Server Name

4. Select the **Auto start** check box.
5. Press the **OK** push button.

Now the VRPG Client GUI Designer knows which AS/400 (server) has the database you need for the next exercise.

1. Use **Define reference fields** from the GUI Designer menu bar to display the field list for file CUSTOML1 in library GUIDES2.
2. Drag-and-drop the following fields on to the subfile part:
 - CUSTNO
 - CUSTNA

Note: If you add the wrong fields, or want to change the order of the fields in the subfile, open the Settings Notebook for the subfile and go to the **Fields** page.

Once you have completed the previous steps, you should have a design window that looks similar to the following figure:

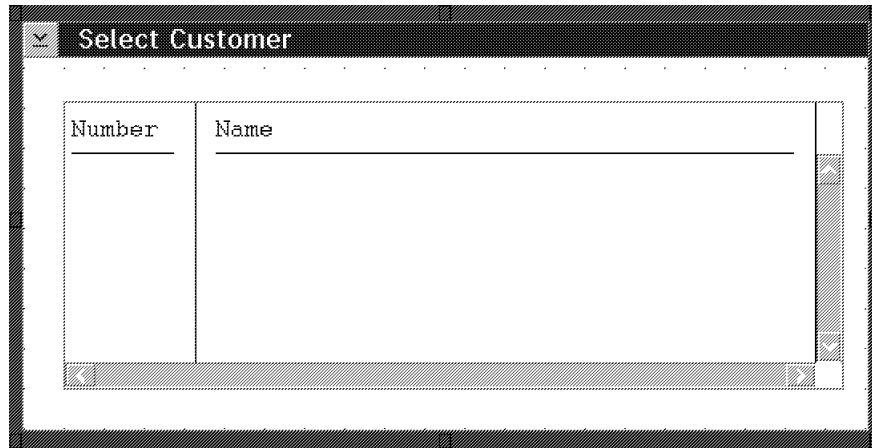


Figure 87. Completed Window with Subfile

Now you need to add code to fill the subfile. In this example, you are going to fill the subfile when the window is created. Therefore, you need to create an Action Subroutine for the windows **CREATE** event.

To create the Action Subroutine, use the right mouse button to invoke the pop-up menu for the window. Choose **Events** and choose **CREATE**.

When the skeleton Action Subroutine is displayed, add the following VRPG Client statements between the **BEGACT** and **ENDACT** statements to fill the subfile:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
CLON01Factor1+++++0pcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
*
C    FRA00014      BEGACT   CREATE      FRA00014
*
* Read a record from database
C          read      customna                      99
*
* Do until end of file
C    *in99          doweq      *off
*
* Add a record to the subfile
C          write      SFL1
*
* Read a record from database
C          read      customna                      99
C          enddo
*
C          ENDACT
*
```

Figure 88. Code to Fill the Subfile from the Database

Save your code.

Now you must add a file specification in your program that is used to fill the subfile.

Adding the File Specification

1. If an LPEX Editor window is not already open, choose **Project** and **Edit source code** from the GUI Designer menu bar.
2. Enter the following File specification for the Customer physical file:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
FFilename++IPEASFRlen+LKlen+AIDevice+.Keywords+++++++
FCustomer  IF      E              K DISK    REMOTE
F                                         Rename(CUSTOM01:CUSTOMNA)
```

Figure 89. "F" Specification for CUSTOMER File

Tip

File specifications must follow any **H** specifications and come before any **D** specifications.

The following values were used to define this "F" specification:

- Customer - The file name used by the program.
- I - File is input only.
- F - File is being processed as full-procedural.
- E - The file is externally described.
- K - The file is being processed by key.
- DISK - This is a disk file.
- REMOTE - The file resides on the AS/400 system.
- RENAME - Rename the record format since it is already used in file CUSTOMLF.

Tip

You might want to add the file keyword **BLOCK(*YES)** in applications that use SETTLL or CHAIN. This allows the RPG compiler to define blocking even if these operation codes are used to access the file.

Selecting a Record from the Subfile

Now you need to add the logic that reads the record selected by the user when the user double-clicks on a subfile record. When the user double-clicks on the subfile part, an **Enter** event is signalled. Therefore, you need to create an Action Subroutine for the subfiles Enter event.

1. From the subfiles pop-up menu, create an Action Subroutine for the Enter event.
2. In this Action Subroutine, use the **READS** operation code to get the selected record from the subfile. The READS moves the value for the customer number into the RPG variable CUSTNO. The READS operation code requires the subfile part name in factor 2 (without the quotes).
3. You must move the customer number to the CUSTNO field on the CUSTINQ window in application Customer Inquiry. Since this field resides in a different component, you must use a component reference part to exchange information.

You can skip this step now and add this function in the next Lab.

4. Finally, you want this subfile window to disappear when a selection has been made. Add the logic required to make this window not visible.

This is all of the logic needed for this Action Subroutine.

Save your code changes by choosing **File** and **Save** from the LPEX Editor menu bar.

Defining AS/400 Information

Before you can build the application, you must indicate which file on the AS/400 system is being used, and on which AS/400 system or Server the file resides. Do this by entering information in the **Define AS/400 information** notebook. The server is still the same as the one used before so no changes are required here.

Defining the AS/400 File

Now you must define the file that is being used to fill the subfile. Display the **Define AS/400 information** window by choosing **Project** and **Define AS/400 information** from the GUI Designer menu bar. Go to the **File** page and press the **Add** push button to add the file CUSTOMER.

The **Override** value is the name we called the file in our program (in this case, **CUSTOMER**).

The **Remote Name** value is the qualified name of the file on the AS/400 system. For this example, the remote file name is **GUIDES2/CUSTOML1**.

The **Server Alias Name** is the alias name of the server that was defined on the **Server** page.

Building the Subfile Component

Now you can build the component. Since the build environment has been changed, you might want to have a look at the build options.

Under the project menu, select **Build options** and OS/2; this brings up the new BUILD options dialog. Specify any changes you want to make in here; you are not prompted at build time for changing the options.

You should note that if you attempt to run this component, it will fail. This is because you have created a VRPG Client component, not an application. VRPG Client components can only be invoked by **starting** them from another VRPG Client component or application. In the following section, you return to the Customer Inquiry application to add logic that starts this subfile component.

Adding a Logic to the Customer Inquiry Component to Start the Subfile Component

Since you want to show the subfile for selecting a customer when the user does not know the customer number, you have to add some capability to the **CUSTINQ** window in application **Customer Inquiry** to do this. Adding a **Find** push button and an Action Subroutine with some logic to start the component **COMPLIST** gets you there.

Select the **Open a Project** Icon on the toolbar.

In the open dialog window, select **Customer Inquiry** and press the **Open** push button.

You are now editing the **Customer Inquiry** application.

1. Open the **CUSTINQ** design window and add a push button part.

2. Change the label of the push button to **Find...**
3. Change the name to **PSBFIND**.
4. Create an Action Subroutine for the **Press** event of this push button to START component **COMPLIST**.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
CLON01Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+HiLoEq
*
C                                START      'COMPLIST'
```

Figure 90. Logic to START the Subfile Component

Save your code changes by choosing **File** and **Save** from the LPEX Editor menu bar, and then build this project.

Running the Enhanced Application

Run the Customer Inquiry application. You now have an additional push button that lists customers from the database. Try it out. When the CUSTINQ window is displayed, press the **Find...** push button to display the list of customers. In a later lab, you see how to retrieve the selected customer number.

Creating Help for Your Application

In this exercise, you create online help for your application. Help in VRPG Client is written for each part. In this example, you create the help that describes the function of the **CUSTNO** Entry Field on the **CUSTINQ** window.

To begin, invoke the pop-up menu for the **CUSTNO** field by clicking it with the right mouse button.

1. On the pop-up menu, choose **Help text**.
2. The LPEX Editor is shown with statements similar to the following example:

```
:h1 res=171.EF00001A
:p.Help
```

The information up to the period on the first statement is used by VRPG Client when invoking help so this information should not be changed.

The string following the period, **EF00001A**, is the heading that is shown in the help window. Change it to something similar to the following example:

```
:h1 res=191.Help for Customer Number Field
```

3. The text on the following line is the body of the help text, and begins with the **Paragraph** tag, (:p.).

Change this text to something similar to the following example:

```
:p.Enter a customer number in this field and press OK.
If the customer number is found, the
:hp9.Customer Info:ehp9.
window is displayed.
To find a valid customer number, press the
:hp9.Find...:ehp9. pushbutton.
```

The :hp9. and :ehp9. tags are called **highlight phrase** tags and cause the text between them to be displayed in a different color.

Tip

A description of all IPF tags can be found in the IPF online reference manual in the VRPG Client folder. You can view this manual by choosing **Help** from the GUI Designer menu bar.

4. Save your help and close the LPEX Editor window.
5. In the GUI Designer menu bar, choose **Project, Build**, and **OS/2** to build this project.

To test your help, invoke the program by selecting the **Run** icon on the Tool Bar. When the **Customer Inquiry** window (CUSTINQ) is shown, move the cursor to the customer number field by clicking on it with the mouse and press the **F1** key. The help you created is shown in a separate help window.

This short example illustrates how easy it is to add online help to your VRPG Client applications. There are many other IPF tags you can use to enhance your help, including adding bitmap images.

Lab Summary

This Lab guided you through:

1. Creating a component
2. Creating a subfile
3. Creating online help

Lab 5: Using the Component Reference Part

Component reference parts are used to monitor for events in other components and to get and set attribute values from parts in other components.

In this lab, you enhance your application using a component reference part to pass the value of the selected customer number from the subfile component **COMPLIST** back to component **Customer Inquiry**.

You are making the following changes:

- For component COMPLIST:
 - Add an entry field part that contains the selected customer number. It is this part that the component reference part is referencing.
- For component Customer Inquiry:
 - Add a component reference part.
 - Change the settings for this part so it monitors for the **CHANGE** event of the entry field added to component COMPLIST.
 - Change the settings for this part so it can retrieve the **Text** attribute for the entry field.
 - Add logic for the **Notify** event of the component reference part to get the changed data.

During this lab, you:

1. Create a hidden field containing the selected customer number.
2. Create a component reference part.
3. Use the component reference part to react to an event and get an attribute value from another component.
4. Write the VRPG Client logic to fill the hidden field with the subfile selection.
5. Write logic to get the customer number value from a different component.
6. Build the application.
7. Run and test the application.
8. Improve it.
9. Test it.

What You Should be Able to Do

As a result of this lab, you can write applications containing multiple components that use component reference parts to communicate between each other by:

- Creating and using component reference parts.
- Writing VRPG Client logic to monitor for events across components.
- Writing VRPG Client logic to get attribute values from other components.
- Learn about the behavior of components.

Creating the Hidden Field

In this Lab, you add function to the subfile application that you started in the previous lab.

The enhancement you make first is to add a hidden entry field to the CUSTLIST window of component **COMPLIST**. The Text attribute of this is set to the customer number value when the user double-clicks on a record in the subfile. The hidden field, in turn, triggers a **CHANGE** event.

The following steps guide you through the process of creating the hidden field in the existing component COMPLIST.

1. Double-click on the Customer Inquiry project on your Desktop.
2. You see all of the parts belonging to the *Customer Inquiry* in the project window including the component **COMPLIST**.
3. Double-click on the **COMPLIST** icon which brings up the project window for this component.
4. Locate the **EDIT** icon on the toolbar and select it. This brings up the GUI Designer.
5. You can now double-click on the CUSTLIST window icon to bring up this window.
6. Select an entry field from the parts palette and put it on the *Custlist* window. Since it is hidden, its position is not important.
7. Change the field name to **EF2**.
8. Deselect the visible check box so it does not appear at run time.
9. Invoke the Pop-up for EF2 and choose EVENTS. From the Events list, choose **CHANGE**. When the editor window opens, you can just close it. The

component reference part requires that an Action Subroutine exists even though it contains no logic.

10. Invoke the pop up for the subfile part **SFL1**.
11. Select **Events** and **Enter**.
12. In the editor for the Enter Action Subroutine, set the **TEXT** attribute of the hidden field EF2 to CUSTNO. This is the customer number from the selected subfile record.

Save your code and build the component. After the component is built successfully, you can now open project Customer Inquiry and make changes to it.

Creating and Using the Component Reference Part

Click on the OPEN icon on the toolbar and select project **Customer Inquiry**.

Now add a component reference part to Window CUSTINQ.

1. Open the CUSTINQ design window.
2. Drag a component reference part from the Parts Palette and drop it on this window.

Open the Settings Notebook for the component reference part and rename it to **MONCHG**. Now go to the **Style** page. The information you need to give a component reference part is twofold:

1. The upper section defines the part attribute being referenced.
2. The lower section defines the part event to monitor.

On the Style page, specify the component name that you want to reference (in this case, **COMPLIST**).

Since you are interested in getting the Value attribute and monitoring for the change event, you must fill out both sections.

1. Specify the reference window name **CUSTLIST**.
2. Specify the reference part name **EF2**.
3. Specify the reference attribute name **TEXT**.
4. This concludes the definition for the upper part of the Style page for this component reference part.
5. Now do the lower part to define the event you want to monitor. In this case, you want to monitor part **EF2** for the **CHANGE** event.
6. Use the same preceding information for window name and part name.
7. Then specify **CHANGE** for the event to be monitored.

The next step involves writing an Action Subroutine that is invoked when component reference part MONCHG gets notified by a CHANGE event from Entry Field EF2. In the VRPG Client component reference part, this is called a **NOTIFY** event.

8. Write an Action Subroutine for the MONCHG Notify event that gets the **ATTRVALUE** value from the MONCHG part and puts it into the variable **CUSTNO**.

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+.
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C      'MONCHG'      Getatr      'AttrValue'      CustNo
```

9. Copy this value to the CUSTNO field in window CUSTINQ with a SETATR statement:

```
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C      'CustNo'      Setatr      CustNo      'Text'
```

10. Save everything you have done.
11. Build the application.
12. Run and test the application. When you double-click on a subfile record, the corresponding customer number should appear in the CUSTNO entry field.

Let's Work on Visibility and Focus

You notice that when the CUSTLIST window is displayed, it does not have focus.

To give this window focus, we use another component reference part in component Customer Inquiry to set the **Visible** and **FOCUS** attributes for window CUSTLIST.

Open the GUI Designer for project Customer Inquiry and open window **CUSTINQ**. Create another component reference part on this window and name it CRP1. Open the Settings Notebook and go to the style page and set:

1. Component to **COMPLIST**.
2. Reference window name to **CUSTLIST**.
3. Reference part name to **CUSTLIST**.

This completes the definition of the component reference part. Now let's add the necessary logic.

First let's add an indicator that tells the application Customer Inquiry that the component COMPLIST is already started and the only thing left to do is to make the window CUSTLIST in component COMPLIST visible and give it focus.

You have to add some code to the Action Subroutine that is linked to the **Find** push button on window **CUSTINQ** in component Customer Inquiry.

1. Use the LPEX Editor and locate the Press event action subroutine for the Find push button.
2. Add code to set on indicator 60 when component COMPLIST is started to indicate that component COMPLIST is started.
3. Add code to skip STARTing component COMPLIST if *IN60 is on.
4. Add the following code to set window CUSTLIST in component COMPLIST visible:

```
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C      'CRP1'      SETATR      'VISIBLE'      'REFATTR'
C      'CRP1'      SETATR      '1'      'ATTRVALUE'
```

5. To give the window focus, add the following two statements:

```
CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++++Len++D+HiLoEq
C   'CRP1'          SETATR   ' FOCUS'      ' REFATTR'
C   'CRP1'          SETATR   ' 1'          ' ATTRVALUE'
```

6. Finally, build and test your application. When the Find push button is pressed, the Subfile window should be visible and have focus.

Lab Summary

This lab guided you through:

1. Creating and using a component reference part.
2. Using the two different functions of a component reference part.
3. Writing VRPG Client logic for a component reference part.

Lab 6: Using Pop-up Menus and Notebooks

In this lab, you enhance the subfile component with a Pop-up menu. The Pop-up menu is shown when the user clicks the right mouse button on the subfile.

To show details of a selected customer, you also add a notebook part.

During this lab, you:

1. Create a Pop-up menu.
2. Create a notebook containing customer information.
3. Write the VRPG Client logic to handle Pop-up menus.
4. Write the VRPG Client logic to fill a notebook page.
5. Run the application.

What You Should Be Able to Do

As a result of this lab, you can write applications containing Pop-up menus and notebooks by:

- Adding a Pop-up menu to a VRPG Client application.
- Creating a notebook in a non-canvas window.
- Writing VRPG Client logic to react to the Pop-up event.
- Writing VRPG Client logic to respond to menu select events.
- Writing VRPG Client logic for notebook parts.

Creating a Pop-up Menu

You are enhancing your subfile component with a Pop-up menu and a notebook. The user can click with the right mouse button on a selected subfile record and your VRPG Client application shows a Pop-up menu. The Pop-up menu gives you the choice of viewing the details of the selected customer or deleting the selected customer record.

The additional enhancement you make pertains to the presentation of the user data. If the user chooses the **Detail** menu item in the Pop-up, you show the customer data in a notebook.

Working with Pop-up Menus:

These steps guide you through the process of creating the Pop-up menu.

1. Double-click on the **COMPLIST** component in your GUIDES2 project window.
2. You see all of the parts belonging to the component **COMPLIST** in the project window.
3. Click on the EDIT button on the toolbar to bring up the **GUI Designer** for this project.
4. Locate the Pop-up icon in the Parts Palette and drag it onto your design window.

The location where you drop it on the window does not matter. At run time, the Pop-up shows where the mouse cursor is located when the right mouse button was clicked.

You notice that the design window is extended at the bottom to accommodate an icon that represents this part. Since the Pop-up part is floating on the window, the GUI Designer keeps a reference icon at the bottom of the window.

5. Name this Pop-up menu **POPUP1** and deselect **VISIBLE**.

Note: The icon at the bottom allows you to define the Pop-up part. The part itself is shown as a menu item that has its own properties. To change the Pop-up part name, you have to use the pop-up reference icon at the bottom of the window. To change the menu item name and react to events from the menu item, you have to use the part itself.

6. Name the Pop-up menu item **POPDETAIL**.
7. Now you have to add one more **menu item** to this Pop-up part.
8. Select a menu item from the parts palette and drop it onto the **POPDETAIL** menu item. Selecting this menu item shows a Notebook with customer information.
9. Name this menu item **POPDELETE**.

Tip

Invoke the Pop-up menu for the Pop-up by first selecting it with the left mouse button and clicking on it with the right mouse button. By selecting **Move**, you can now move the Pop-up menu items next to the Pop-up menu part.

Note: The Pop-up part is not supported for Windows.

Adding VRPG Client logic to support Pop-up menus.

1. Select the **POPUP** event on subfile SFL1.
2. Add the following code to the action subroutine for the **POPUP** event:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
CLON01Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+HiLoEq
C      'POPUP1'      SETATR      1      'VISIBLE'
```

Figure 91. Displaying the Pop-up Menu

This piece of code tells the run time to make the Pop-up part visible. Pop-up menus are always displayed at the mouse cursor position.

3. Close this window.

Creating the Notebook

You create a notebook with two pages to show the customer data of the selected customer record in the subfile. One page contains name and address information; the other page contains contact information.

To introduce the notion of a **Window without Canvas**, you are creating a notebook that fills the entire window.

1. Go to the Parts Catalog and locate the **Window** part. It is in the **Frames** page.
2. Drag it to the Project View of the Graphical User Interface window.
3. Name this window **WINNOTE**.
4. Change the title to **Customer Detail Notebook**.
5. Drag-and-drop a notebook icon onto the WINNOTE window.

Using a Window without Canvas causes the notebook to fill the entire window frame.

6. Name this notebook **NBKCUST** and deselect **VISIBLE**.
7. Drag-and-drop two **Notebook page with canvas** parts onto the notebook.

Tip

You can reorder notebook pages by opening the Settings Notebook for the notebook part and going to the **Page List** page.

8. Open the Settings Notebook for each page; change the Tab type to Minor tabs.
9. Change the tab text on each of the pages to:
 Address
 Contact
10. Now use the database reference function to create the following fields on the address page:
 Customer number
 Address
 City
 Phone
11. Get the **CUSTNO** and **Contact** field and drag-and-drop them to the **Contact** notebook page.

Note: The CUSTNO field on the second page gets a new name. Name scoping is on a window base. Only one unique name per window per part is allowed, so VRPG Client renamed this field when it was placed on the window.
12. Save your project.

Add Logic to Update the Notebook Pages

There is not a lot involved to write the logic for this piece of the application.

1. Write an action subroutine that reacts to the MenuSelect event of the **POPDetail** menu part.
2. Get the selected record from the subfile (**READS** operation code).
3. Use the **CUSTNO** field to chain to the AS/400 file.
4. WRITE to window **WINNOTE**.
5. Make window **WINNOTE** visible and set focus.

Tip

Do not forget to set the text attribute for the customer number on the Contact page. Check the manuals to determine whether to use setatr or %setatr.

If you have time, you might want to implement the delete function for a customer record as well by creating an action subroutine for the MenuSelect event of the **POPDELETE** menu item and delete the selected subfile record.

You also might want to add some code to make the WINNOTE window invisible if the close event is received.

Tip

You can specify on Factor 2 of the ENDACT operation code whether **Default processing** should take place or not. By setting this value to ***NODEFAULT**, you can prevent the window from actually closing.

This is useful if the end user double-clicks on the system icon of a window and you want to prevent the window from being closed.

6. Save and build your application.
7. Run and test your application.

Note: The notebook part and Pop-up menu part are not supported in Windows.

Lab Summary

This Lab guided you through:

1. Creating a Pop-up menu.
2. Creating a notebook with notebook pages.
3. Writing logic to deal with these parts.

Lab 7: Using the Container Part

In this lab, you create an application that uses the container part. The application also reads records from a local file to add records to the container. During this lab, you:

1. Create a container part and add columns.
2. Write the VRPG Client logic to read records from a local file.
3. Write the VRPG Client logic to add records to the container part.
4. Write the VRPG Client logic to change the container view.

5. Build the application.
6. Run the application.

What You Should Be Able to Do

As a result of this lab, you can write applications that use program described files to access local files and also use the container part by:

- Creating containers with multiple columns.
- Writing VRPG Client logic to add records to a container.
- Writing VRPG Client logic to change a container view.
- Writing VRPG Client logic to read records from a local file.

Container Overview

The container part allows you to display records in three different views:

- Tree view
- Icon view
- Details view

In addition, records are added in a hierarchical manner (that is, a record can have one or more sub-records). A record that has sub-records is called the **Parent** record. Records added to this record are referred to as **Child** records.

Application Description

In this exercise, you use a container part that displays records stored in a database on the workstation. The database is a list of employee records and has the following record layout:

1 - 3	Department number
5 - 5	Employee type
7 - 29	Name
30 - 42	Phone

If column 1 contains an asterisk (*), it is considered a comment record and should be ignored.

If the **Employee type** field is an **M**, the record is considered a manager record and, therefore, has a **ParentID** value of **0**. An Employee type of **E** indicates this employee works for a manager and should be added as a child record to the appropriate manager record using the department number as the **ParentID**.

Creating the Graphical User Interface (GUI)

If the GUI Designer is currently active, choose **Project** and **New** to create a new application.

Perform the following steps to create the container part:

1. Use direct editing or the window Settings Notebook to change the title of the window to something appropriate such as **Departments**.
2. Open the Settings Notebook for the window, go to the **Style** page, and select **Sizable** as the **Border** style.
3. Close the windows Settings Notebook.
4. For this lab, we want the container part to always fill the window part. Therefore, we need to delete the canvas part from the initial design window.

- Invoke the Pop-up menu for the **Canvas** part and choose **Cut** to delete the Canvas part.
5. Locate the **Container** part and drag-and-drop it onto the design window. Notice that the container completely fills the window frame.
 6. Open the Settings Notebook for the Container part by double-clicking on it.
 7. Set the name to CT1.
 8. Select the **Style** tab to go to the Style page. On the Style page, select **Details** as the view. Also, deselect the **Visible** check box for the title.
 9. Now you define the columns for the container. Select the **Columns** tab to go to the Columns page and press the **Add after...** push button.
 10. On the **Add column** window, select **Object Icon Column** as the type.
 11. Change the width value to 64 pixels and select **Center** for **Horizontal alignment**.
 12. Press the **OK** push button to add this column and return to the column page.
 13. Press the **Add after...** push button to add a second column to the container.
 14. On the Add column window, select **Object Text Column** as the column type. Change the width value to 200 pixels.
 15. In the **Heading definition** section, type **Name** as the **Text** value.
 16. Press the **OK** push button to add this column to the container and return to the Column page.
 17. Press the Add after... push button to add a third column to the container. This column contains the phone number.
 18. On the Add column window, select **Text column** as the column type and change the width value to 200 pixels.
 19. Press the **OK** push button to add this column to the container and return to the column page.
 20. Add the code for the pop-up event.
 21. Close the Settings Notebook for the container part.

Defining the Local File

Since the database to be read is on the workstation, you must define a program described file in your program. Open an Editor session by choosing **Project, Edit source code** from the GUI Designer menu bar, and add the following **File** specification:

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+.
FFilename++IPEASFRlen+LKlen+AIDevice+.Keywords+++++++
FContacts IF F 256 DISK USROPN EXTFILE(file)
F RCDLEN(Field1)
DField1 S 5 0 INZ(256)
*
```

Figure 92. Defining the Local File

The **USROPN** keyword indicates that this file is opened at program execution. The **EXTFILE** keyword specifies that the field named **file** contains the actual file name to be processed.

Defining the Input Specifications

Now you must define the record layout for the employee record department file using Input specifications. The first record definition is for comment records and the second is for employee records.

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+.
IFilename++SqNORiPos1+NCCPos2+NCCPos3+NCC.....
IContacts NS 02 1 C*
*
IContacts NS 01
I              1 3 DeptA
I              1 3 0DeptN
I              5 5 Type
I              7 29 Name
I              30 42 Phone
```

Figure 93. Defining the Record Types

Defining Program Variables

Add the following **D** specifications to your program. These specifications define variables and constants that are used by the program.

```
*...1....+...2....+...3....+...4....+...5....+...6....+...7....+.
DName+++++++ETDsFrom+++To/L+++IDc.Keywords+++++++
*
* Define constants used by the program
DManager      C          'M'
DEmployee      C          'E'
DMgrIcon       C          '.\MANAGER.ICO'
DEmpIcon       C          '.\EMPLOYEE.ICO'
*
* Define variables used by the program
DFile          S          256   inz('D:\CONTACTS.TXT')
DRecordID      S          6 0
DParent        S          6
DRecord        S          64
DIconFile      S          64
```

Figure 94. Defining Definition Specifications

Note for Icon Files

The two icon files need to be in your RT_OS2 directory. Copy them from directory LAB7 if that has been loaded on your PC, or copy them from the diskette.

The contacts.txt file should be in your D drive. If not, change the drive letter in your D spec. Copy contacts.txt from directory LAB7 or from the diskette.

Adding Records to the Container

To read records from the workstation database to add to the container, use the **CREATE** event for the window part. Create an action subroutine for the window's **Create** event by invoking the Pop-up menu for the window part and choose **Events** and **CREATE**.

Type the following logic for the Create event:

```

*...1....+...2....+...3....+...4....+...5....+...6....+...7....+.
CLON01Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+HiLoEq
*
C      FRA0000B      BEGACT      CREATE      FRA0000B
* Open employee file and get the first record
C      Open      Contacts
C      Read      Contacts      99
* Process all records in the employee file
C      DoW      *IN99 = *OFF
* Ignore comment records
C      If      *in01 = *on
* Handle Manager records
C      If      Type = Manager
C      Eval      RecordIDC = DeptA
C      Eval      IconFile = MgrIcon
C      Eval      Parent = '0'
C      Eval      Name = %Trim(Name) + '(' + DeptA + ')'
* Handle Employee records
C      Else
C      'CT1'      GetAtr      'GetNewID'      RecordID
C      Eval      Parent = DeptA
C      Move      RecordID      RecordIDC      6
C      Eval      IconFile = EmpIcon
C      EndIf
* Since AddRcd uses spaces as a delimiter, change spaces
* to underscores
C      'CT1'      Xlate      Name      Name
C      Eval      Record = RecordIDC + ' ' +
C      %Trim(Name) + ' ' +
C      %Trim(IconFile) + ' ' +
C      %Trim(Parent) + ' ' +
C      Phone
C      'CT1'      Setatr      Record      'AddRcd'
C      EndIf
C      Setoff      0102
C      Read      Contacts      99
C      EndDo
*
C      Close      Contacts
C      Seton      LR
*
C      ENDACT
*

```

Figure 95. Logic to Add Records to the Container

Changing the Container View

To allow the user to change the container view, add a **Pop-up** menu part to the design window. This Pop-up menu is displayed when the user presses the right mouse button when the mouse pointer is over the container.

To create the Pop-up menu:

1. Locate the Pop-up menu part on the Parts Palette and drag-and-drop it onto the design window. This also creates the initial menu item.
2. Drag-and-drop two more menu items from the Parts Palette onto the first menu item.

3. Use the Settings Notebook for the menu items, or use direct editing to change the text of the menu items to one of the following views:
 - Tree View
 - Icon View
 - Details View
4. For each menu item, create an action subroutine to change the view of the container by setting the **View** attribute of the container. Refer to the **Parts Reference** manual, which is online and can be invoked from the LPEX Editor or GUI Designer menu bar, to see what values the **View** attribute must be set to for each view.

Testing the Container Application

Compile and run the container application. Change the container view by using the Pop-up menu and notice the difference in the views. Notice that the **Details** view is the only view that displays all of the records.

Also notice, that in the **Icon** view, the record icon is not always visible. To fix this, you need to add a statement in the **Icon View** Action Subroutine to change the container's **Arrange** attribute. Again, refer to the Parts Reference manual to see how this is done. Make this change to your program, recompile, and build to test your changes.

This completes the Container lab.

Hands On Lab Summary: This concludes the VRPG Client hands-on lab. We hope you enjoyed the experience. You can now build on this experience and create your own VRPG Client applications by exploring many of the other features of VRPG Client that were not covered in this lab.

Have fun with VRPG Client!

Descriptions of Files Used in the Labs

The following sections contain the description of files used during the hands on Lab.

DDS for Customer File

The following DDS example is for the file CUSTOMER/CUSTOM01 in library GUIDES2.

```

...A*. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+ ...7
***** Beginning of data *****
A* CELDIAL CUSTOMER FILE
A          R CUSTOM01
A          CUSTNO          7          COLHDG('Customer')
A          CUSTNA          40         COLHDG('Company name')
A          REPNO           5          COLHDG('Rep identifier')
A          CONTAC          30         COLHDG('Name')
A          CPHONE          17         COLHDG('Telephone')
A          CFAX            17         COLHDG('Fax')
A          CADDR           40         COLHDG('Address')
A          CCITY           30         COLHDG('City')
A          CCOUNT          20         COLHDG('Country')
A          CZIP            10         COLHDG('Postal Code')
A          CZIPLO          1          COLHDG('PC location;')
A                                     VALUES('1' '2' '3')

```

Figure 96. DDS for Customer File

The following DDS example is for the file CUSTOMLF/CUSTOM01 in library GUIDES2.

```

...A*. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+ ...7
***** Beginning of data *****
A          R CUSTOM01          PFILE(GUIDES2/CUSTOMER)
A          K CUSTNO

```

Figure 97. DDS for Logical File CUSTOMLF

The following DDS example is for the file CUSTOML3/CUSTOM01 in library GUIDES2.

```

...A*. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+ ...7
***** Beginning of data *****
A          R CUSTOM01          PFILE(GUIDES2/CUSTOMER)
A          K CUSTNO

```

Figure 98. DDS for Logical File CUSTOML3

Customer File on AS/400 System

The following example is a partial sample of fields and data from file CUSTOML3/CUSTOM01 in library GUIDES2. The instructor can use the data files from the diskette, create the files and enter sample data from the following example, or come up with unique data. The file on the diskette contains 72 records.

Customer number	Company name	Name	Telephone	Address	Postal Code	City	PC Loc
0010100	Meridien Electr	Alfredo Bayonne	206-865-4027	10423 S.E. 30th	98007	Bellevue, WA	
0010200	Royal Hardware	Arnie Podell	905-619-2045	Maple View Plaz	L8D 4S6	Ajax, Ontario	
0010300	Webster Applian	Bob Wolfstadt	619-549-5212	7350 Miramar Ro	92121	San Diego, CA	
0010400	ProLine Buildin	Bud Dobbs	905-403-4055	73 Marchwood Ro	D8G 3V6	Burlington, Ontario	
0010500	Donnamora Const	Byron Goeds	905-805-2295	Woodbridge Plaz	F7V 5S7	Woodbridge, Ontario	
0010700	Universal Commu	Dave Franken	415-545-5055	720 Harrison St	94107	San Francisco, CA	
0010800	Baker Electroni	Dave Matthison	818-715-2045	17150 Koll Cent	94566	Pasadena, CA	
0010900	Village Telepho	Dayle Swigger	707-367-4530	2752 Betsy Ross	95054	Santa Clara, CA	
0011000	Bayview Direct	Claus Weiss	416-448-3987	201 Bloor Stree	M8B 7F5	Toronto, Ontario	
0011100	BelAir Communic	Doreen Coin	914-765-8021	302 Washington	71530	White Plains, NY	
0011300	Burnham Trading	Efrem Helassie	613-225-0753	91 Baseline Roa	C6B 9S3	Trenton, Ontario	
0011400	Calderone Impor	Elsie Pons	407-392-7077	702 S.W. 15th S	33486	Boca Raton, FL	
0011500	The Communicati	Esther Varrick	904-599-0377	3011 E. Georgia	33830	Jacksonville, FL	
0011600	Sudbury Radio a	Garry Morehouse	705-522-5044	8 North Road	P7G 5A3	Sudbury, Ontario	
0011700	Christies Elect	George Baccxrat	818-707-6767	70223 Agoura Ro	91354	Westlake Village, CA	
0011900	Conroy Communic	Guy Lewis	918-825-4545	701 S. Adair St	74361	Pryor, OK	

Figure 99. Data in CUSTOML3 File

Contacts.TXT File

The following example contains the layout of the CONTACTS.TXT file used in LAB7.

```
* Department 817
817 M Eli Javier          (416)448-9999
817 E Larry Schweyer     (416)448-3344
817 E Hans Koert         (416)448-3411
817 E Phil Coulthard     (416)448-2199
*
* Department 814
814 M Enoch Ng           (416)448-9944
814 E Alan Chao           (416)448-7733
814 E Larry Keeling      (416)448-8831
814 E Bobby Siu           (416)448-9900
814 E Paul Kao           (416)448-3388
*
* Department 059
059 M Mark Changfoot     (416)448-0311
059 E Brian Farn         (416)448-9931
059 E Andrew Kerr        (416)448-9911
059 E Derek Lewsey       (416)448-2711
059 E Vincent Suen       (416)448-3992
*
* Department 546
546 M George Farr        (416)448-9221
546 E Dave Cheng         (416)448-9211
546 E Sarah Ettritch     (416)448-8211
546 E John Fellner       (416)448-8222
546 E Scott Ripley       (416)448-2234
```

Figure 100. Data in CONTACT.TXT File

VRPG Source GUIDES2

The following example contains the final VRPG Client application from Lab 1 through Lab 6.

```
*****
*
* Program ID . . . :
*
* Description . . : Sample VRPG program to demonstrated the Component*
*                   Reference part.
*
*                   When the Find push button is pressed on this
*                   component, a second component, COMPLIST is
*                   invoked which displays a subfile containing all
*                   customer records. When the user double-clicks
*                   on a customer record, the customer number is
*                   set into the CUSTNO entry field in this component*
*
* Function . . . :
*
* Messages . . . :
*
* Files . . . . :
*
* Input . . . . :
*
* Output . . . . :
*
* Change activity:
*
* Who  Date   Flag  Description
* ---  -----  ---  -
*
*****
*
H
Fcustomlf if E          k disk  remote
Dinfo box          m          style(*info)
D                    button(*retry: *abort: *enter)
*
Dmsg1              m          msgnbr(*msg001)
*
Dmsg2              m          msgnbr(*msg002)
D                    msgdata(custno)
*
Dwarn box          m          style(*warn)
D                    button(*yesbutton: *nobutton)
*
Dtext              m          msgtext('Number Not Found')
```

```

*****
*
* Window . . : *
*
* Part . . . : *
*
* Event . . : *
*
* Description: *
*
* Change activity: *
*
* Who Date Flag Description *
* --- ----- *
*
*****
*
C PSBEXIT BEGACT PRESS CUSTINQ
C move *on *inlr
*
C ENDACT
*****
*
* Window . . : *
*
* Part . . . : *
*
* Event . . : *
*
* Description: *
*
* Change activity: *
*
* Who Date Flag Description *
* --- ----- *
*
*****
C PSBOK BEGACT PRESS CUSTINQ
C 'psbok' getatr 'Backcolor' bcolor 2 0
C if bcolor <> *RED
C 'psbok' setatr *RED 'backcolor'
C else
C 'psbok' setatr *PALEGRAY 'Backcolor'
C end
C 'custno' getatr 'text' custno
* read the customer file, if no customer found

```

```

C      custno      chain      customlf      50
C      if          *in50 = *off
C      showwin     'custinfo'      51
C      if          *in51 = *on
C      eval        %setatr('custinfo':'custinfo':'visible')=1
C      eval        %setatr('custinfo':'custinfo':'focus')=1
C      endif
C      write       'custinfo'
C      else
C      msg1        DSPLY      infobox      reply      9 0
C      'custno'    setatr      *red        'backcolor'
C      endif
C      *
C      ENDACT
*****
*
* Window . . :
*
* Part . . . :
*
* Event . . :
*
* Description:
*
* Change activity:
*
* Who   Date   Flag  Description
* ---  -----  ---  -
*
*****
C      INFOOK      BEGACT      PRESS      CUSTINFO
C      *
C      'CUSTINFO'  setatr      0          'visible'
C      ENDACT
*****
*
* Window . . :
*
* Part . . . :
*
* Event . . :
*
* Description:
*
* Change activity:
*
* Who   Date   Flag  Description
* ---  -----  ---  -
*

```



```

*****
*
C      CUSTINQ      BEGACT      ACTIVATE      CUSTINQ
c      'CUSTINQ'    setatr      1              'focus'
c      'CUSTNO'     setatr      1              'focus'
*
C                                  ENDACT
*****
*
* Window . . . :
*
* Part . . . :
*
* Event . . . :
*
* Description:
*
* Change activity:
*
* Who  Date   Flag  Description
* ---  -----  ---  -----
*
*****

C      PSBFIND      BEGACT      PRESS          CUSTINQ
*      put up the customer list window, if the window is all ready up
*      just set focus to it.
C                                  start      'complist'          65
C                                  if          *in65 = *on
C                                  eval        %setatr('custing': 'custing': 'focus')=0
C                                  endif
C                                  ENDACT
*****
*
* Window . . . : CUSTINQ
*
* Part . . . :
*
* Event . . . : NOTIFY
*
* Description:
*
* Change activity:
*
* Who  Date   Flag  Description
* ---  -----  ---  -----
*
*****

C      MONCHG      BEGACT      NOTIFY          CUSTINQ
C      'MONCHG'     GETATR      'attrvalue'     CUSTNO
C      'CUSTNO'     SETATR      CUSTNO          'TEXT'
C                                  ENDACT

```

Figure 101 (Part 4 of 4). Customer Inquiry Program

VRPG Source for COMPLIST

The following example contains the customer list component.

```
FCustomer IF F 186      DISK    usroprn extfile(file)
*
DFile          S          15      Inz('.\CUSTOMER.DAT')
*
ICustomer NS 01
I              1   7  CustNo
I              8  47 CustNa
I             53  82 Contac
I             83  99 CPhone
I            117 156 CAddr
I            157 186 CAddr2
*****
*
* Window . . :
*
* Part . . . :
*
* Event . . :
*
* Description:
*
*****
*
C      CUSTLIST      BEGACT      CREATE      CUSTLIST
*
C              Open      Customer
C              Read      Customer          99
*
C      *in99      DowEq      *off
C              Write      SFL1
C              Read      Customer          99
C              EndDo
*
C              Close      Customer
*
C              EndAct
*****
*
* Window . . :
*
* Part . . . :
*
* Event . . :
*
* Description:
*
*****
```

```

C      SFL1          BEGACT   ENTER      CUSTLIST
*
C      ReadS         SFL1
C      'EF2'         Setatr   CustNo
C*     'CustList'    Setatr   0          'Visible'
*
C                               EndAct
*****
*
* Window . . :
*
* Part . . . :
*
* Event . . :
*
* Description:
*
*****
*
C      EF2           BEGACT   CHANGE      CUSTLIST
*
*
C                               EndAct
*****
*
* Window . . :
*
* Part . . . :
*
* Event . . :
*
* Description:
*
* Change activity:
*
* Who   Date    Flag  Description
* ---  -
*****
*
C      SFL1          BEGACT   POPUP       CUSTLIST
*
C      'POPUP1'      Setatr   1          'Visible'
*
C                               ENDACT

```

91

```

*
*****
*
* Window . . :
*
* Part . . . :
*
* Event . . :
*
* Description:
*
*
*****
*
C    PU_DETAIL    BEGACT    MENUSELECT    CUSTLIST
*
C          Reads    SFL1
C          Move1    CustNo      Tmp          7
C          Open    Customer
C          Read    Customer          99
C          DoW    CUSTNO <> Tmp
C          Read    Customer          99
C          EndDo
C          ShowWin 'Detail'
C          Write  'Detail'
C          Close   Customer
*
C          ENDACT
*****
*
* Window . . :
*
* Part . . . :
*
* Event . . :
*
* Description:
*
*
*****
C    CUSTLIST    BEGACT    CLOSE    CUSTLIST
*
C          Move    *on      *inlr
*
C          ENDACT
*****
*
* Program ID . . :
*
* Description . . :
*
*
*****
*

```

Figure 102 (Part 3 of 3). Customer List Program

VRPG Source for Container Example

The following example contains the container application for Lab 7.

```
*****
*
* Program ID . . : CONTAIN
*
* Description . . : VRPG Container example
*
*****
*
* Define the employee record file
FContacts IF F 256 DISK usroprn extfile(file) Employee file
F RCDLEN(Field1)
DField1 S 5 0 INZ(256)
*
* Define constants used by the program
DManager C 'M' Manager record
DEmployee C 'E' Employee record
DMgrIcon C '.\\MANAGER.ICO' Manager icon file
DEmpIcon C '.\\EMPLOYEE.ICO' Employee icon file
*
* Define variables used by the program
DFile S 12 inz('CONTACTS.TXT') File to open
DRecordID S 6 0 Container record ID
DParent S 6 Parent ID
DRecord S 64 Container record
DIconFile S 64 Icon file name
*
*
IContacts NS 02 1 C* Comment record
*
* Record layout for employee file
IContacts NS 01
I 1 3 DeptA Department - char
I 1 3 0DeptN Department - num
I 5 5 Type Employee type
I 7 29 Name Name
I 30 42 Phone Phone number
*
*****
*
* Window . . : FRA0000B
*
* Part . . . : FRA0000B
*
* Event . . : Create
*
* Description: Add records from the employee database to the
* Container part when the window is created. *
*
*****
```

```

C      FRA0000B      BEGACT      CREATE      FRA0000B
*
* Open employee file and get the first record
C      Eval      File = %Trim(File)
C      Open      Contacts
C      Read      Contacts
C      *
C      * Process all records in the employee file
C      DoW      *IN99 = *OFF
C      *
C      * Ignore comment records
C      If      *in01 = *on
C      *
C      * Handle Manager records
C      If      Type = Manager
C      Eval      RecordIDC = DeptA
C      Eval      IconFile = MgrIcon
C      Eval      Parent = '0'
C      Eval      Name = %Trim(Name) + '(' + DeptA + ')'
C      *
C      * Handle Employee records
C      Else
C      'CT1'      GetAtr      'GetNewID'      RecordID
C      Eval      Parent = DeptA
C      Move      RecordID      RecordIDC      6
C      Eval      IconFile = EmpIcon
C      EndIf
C      *
C      * Since AddRcd uses spaces as a delimiter, change spaces
C      * to underscores
C      ' ':_      Xlate      Name      Name
C      Eval      Record = RecordIDC + ' ' +
C      %Trim(Name) + ' ' +
C      %Trim(IconFile) + ' ' +
C      %Trim(Parent) + ' ' +
C      Phone
C      'CT1'      Setatr      Record      'AddRcd'
C      EndIf
C      *
C      Setoff
C      Read      Contacts
C      EndDo
C      *
C      Close      Contacts
C      *
C      ENDACT

```

Open file
 Read first record
 Do until EOF
 If not comment
 Manager record?
 Use Dept as ID
 Set icon file name
 Parent record
 Employee record
 Get a new ID
 Set parent ID
 Make it character
 Set icon file name
 Convert spaces
 Construct new record
 Add the record
 End-not comment
 0102
 99 Get next record
 End-Not EOF
 Close file

```

*****
*
* Window . . : FRA0000B
*
* Part . . . : MI_TREE
*
* Event . . : MenuSelect
*
* Description: Change the Container view to Tree view
*
*****
C      MI_TREE      BEGACT      MENUSELECT      FRA0000B
*
C      'CT1'        Setatr    2          'View'          Set to Tree view
*
C      ENDACT
*****
*
* Window . . : FRA0000B
*
* Part . . . : MI_ICON
*
* Event . . : MenuSelect
*
* Description: Change the Container view to Icon view
*
*****
C      MI_ICON      BEGACT      MENUSELECT      FRA0000B
*
C      'CT1'        Setatr    1          'View'          Set to Icon view
C      'CT1'        Setatr    1          'Arrange'        Arrange icons
*
C      ENDACT
*****
*
* Window . . : FRA0000B
*
* Part . . . : MI_DETAIL
*
* Event . . : MenuSelect
*
* Description: Change the Container view to Details view
*
*****
C      MI_DETAIL    BEGACT      MENUSELECT      FRA0000B
*
C      'ctl'        setatr    3          'view'          Set to Details view
*
C      ENDACT
*****
*
* Window . . : FRA0000B
*
* Part . . . : CT1
*
* Event . . : PopUp
*
* Description: Display the popup menu for the container part
*
*****
C      CT1          BEGACT      POPUP          FRA0000B
*
C      'PMN00010'   Setatr    1          'Visible'       Make popup visible
*
C      ENDACT

```

Figure 103 (Part 4 of 4). Container Program

README File

This is the README.TXT file that is also part of the 4841set.EXE file on the companion diskette.

Setup

To run the labs, you should be equipped with OS/2 workstations running OS/2 WARP or later that are connected to an AS/400 system running OS/400 V3R1 or V3R6. The AS/400 connection may be through Communications Manager/2 (CM/2), or Client Access for OS/2.

The AS/400 system must have ADTS installed. If the AS/400 system is at V3R1, PTF SF25794 must be applied. If the AS/400 system is at V3R6, PTF SF26792 must be applied.

The latest version of VRPG Client, 5763VR1 or 5716VR1, must be installed on all workstations.

These workstations must meet the minimum VRPG Client requirements:

- OS/2 Warp V3 or later
- 16MB of memory
- 20MB of swapper
- 486/33 CPU
- CM/2 or Client Access for OS/2

The lab sections refer to data on the AS/400 system. This data is in a library called GUIDES2. Ensure that this library is installed before running the labs.

Installation and Setup

To copy the solution and directories to your workstation:

1. Run the self-extracting files on the diskette by typing the following command from an OS/2 command line:

4841set X:\ /p

where X is the drive you want to create directory *4841set* on and copy the files to.

2. Use the same approach to extract the other files:

a. UPSAVF X: /P

b. GUIDES2 X: /P

c. COMPLIST X: /P

d. CONTAINR X: /P

e. LAB7 X: /P

f. 4841FLW X: /P

3. Use the CRTPRJ command to create the solution and upsavf directories on your workstation. The CRTPRJ command has the following syntax:

CRTPRJ /create target title

where */create* indicates the project files are already on the workstation, *target* is the target directory, and *title* is the title string for the project. The

title string can be any number of words, and it appears next to the project folder on your OS/2 desktop when the folder is created. For example, type:

CRTPRJ /create x:\UPSAVF Upload files

on an OS/2 command line to create project UPSAVF with project files located in directory **x:\UPSAVF** on your hard drive.

After you use the CRTPRJ command, the VRPG Client/2 project is shown on your desktop.

Note:

For instructions on how to use the program **UPSAVF** to create the AS/400 Lab environment, see: "Help for AS/400 Upload Program" on page 206.

4. Use the CRTPRJ command for directories **UPSAVF**, **CONTAINR**, **GUIDES2**, and **COMPLIST**.
5. Later on during the Lab exercises, make sure when you are doing Lab 7 that you specify the correct drive to find **contacts.txt**.
6. The Freelance files can be used with the SHOW command which is part of the packaged files, so you do not need to own the Freelance product to view these files.

Contents of the Diskette

The diskette contains the following:

File 4841SET.EXE

Run **4841set.exe** to create these files in directory 4841SET:

- README.TXT file with instructions on how to use the files on the diskette.
- CRTPRJ.CMD to create VRPG Client projects from an OS/2 command line.

File UPSAVF.EXE

Run **UPSAVF.EXE x:\ /P** to create these files in directory UPSAVF. The UPSAVF VRPG application allows you to create the AS/400 environment for the hands on LABS in Part 2 of this Redbook:

- The UPSAVF VRPG project files
- PF Customer
- Source PF QDDSSRC - DDS for customer file (physical and logical)

File GUIDES2.EXE

Run **GUIDES.EXE x:\ /P** to create project files in directory GUIDES2. It contains the source and runtime solution for Lab 1 through Lab 6.

File COMPLIST.EXE

Run **COMPLIST.EXE x:\ /p** to create project files in directory GUIDES2. It contains additional source solutions for Lab 1 through Lab 6.

File CONTAINR.EXE

Run **CONTAINR.EXE x:\ /p** to create project files in directory CONTAINR. It contains the source and run-time solution for Lab 7.

File 4841FLW.EXE

Run **4841FLW.EXE x:\ /p** to create files for two Freelance presentations in directory 4841FLW. Use the SHOW command to view these presentations. The SHOW command is part of this package.

- **OVERV04:**

VRPG Client, an introduction to Visual Programming with VRPG Client.

A set of foils to use as an overview and intro to the labs.

- **4841:**

Overview of installation and configuration.

File LAB7.EXE

Run **LAB7.EXE x:\ /P** to create directory LAB7. It contains the local PC files and icons used in Lab 7.

- Contacts.txt
- Manager.ico
- Employee.ico

Help for AS/400 Upload Program

This section contains the help text for the AS/400 upload program.

Note:

Before you do these steps you have to follow the instructions to self extract the UPSAVF project files and use the CRTPRJ command to create a VRPG project.

See also "Installation and Setup" on page 204.

The program **UPSAVF** uploads the DDS and data files required for the VRPG Client labs. It creates the following files in the library that has already been created on the AS/400 system.

- QDDSSRC - Contains DDS source for creating the physical and logical files.
- CUSTOMER - This physical file contains customer records.

Note: Before the **UPSAVF** program is executed, the .RST file must be modified to reference the correct AS/400 system. To do this, use the following steps:

- Go to the project's run-time directory **RT_OS2**.
- Run the following command to "unhide" the .RST file:

```
ATTRIB -h *.*
```

- Use an editor to edit the .RST file.
- Change the value of the **REMOTE_LOCATION_NAME** to reflect the name of the AS/400 system.
- Save the .RST file.
- Copy the run-time **RT_OS2** directory to the **TEST** directory in the **CODE** root directory. This has to be done because you are working on a development workstation and all VRPG application run-time DLL's are expected to reside in this **TEST** directory. Use the command:

```
xcopy x:\UPSAVF\RT_OS2\*. * x:\CODE\TEST\
```

where x: indicates the drive letter these directories are located on.

To upload the lab files, do the following:

- On the AS/400 system, create a library to receive the files. For the VRPG Client lab, this library should be named **GUIDES2**.
- Run the **UPSAVF** program (click with the right mouse button on project **UPSAVF** and select **RUN** from the POPUP menu).
- In the Window that is shown, type library name **GUIDES2** in the **Library** field if it is not already **GUIDES2**.
- Press the **Upload** push button to begin the upload. Do not select any of the check boxes. They are used as status indicators.
- As each step completes, the corresponding check box is checked.
- When the upload is complete as indicated by the **Done** check box, go to the AS/400 system and compile the two logical files.

Source for AS/400 upload program

The following example contains a sample of VRPG Client source code that can be used to upload the AS/400 files for the labs.

```
*****
*
* Program ID . . :
*
* Description . : This program will upload the DDS and data
*                  file required for the VRPG Client lab. It
*                  will put up a window for the user to enter
*                  the AS/400 library which has been created
*                  to contain the lab files. GUIDES2 is the
*                  library which should be created.
*                  Files QDDSSRC - contains DDS source
*                  CUSTOMER- This physical file contains
*                  customer records
*****
*
FDD SIN      IF  F 128      Disk  EXTFILE(PCFILE) usroprn rcdlen(Rec1n1)
FPCCUST      IF  F 217      Disk  EXTFILE(cust) usroprn rcdlen(Rec1n2)
FQDDSSRC     0   E          Disk  Remote usroprn Rename(QDDSSRC:FMT01)
FCUSTOMER    0   E          Disk  Remote usroprn Rename(CUSTOM01:FMT02)
*
DRcd1n1      S              5  0 INZ(128)
DRcd1n2      S              5  0 INZ(217)
DSaveFile    S              14A
DCMD          S             128A
DMEBER        S             10A
DCUSTFILE     S             21A  INZ(' TESTLIB/CUSTOMER')
DTCFILE       S             21A  INZ(' TESTLIB/QDDSSRC')
DPCFILE       S             15A
DCUST         S             15A  INZ(' . \\CUSTOMER.DAT')
DQCMDDDM      S             7A   INZ(' QCMDMDM') LINKAGE(*SERVER)
DCMDLEN       S            15P 5  INZ(%SIZE(CMD))
DLC           S             26   INZ(' abcdefghijklmnopqrstuvwxyz')
DUC           S             26   INZ(' ABCDEFGHIJKLMNOPQRSTUVWXYZ')
*
IDDSIN       NS  01
I              1   80  SRCDTA
```

IPCCust NS 01

I	1	7	CustNo
I	8	47	CustNa
I	48	52	RepNo
I	53	82	Contac
I	83	99	CPhone
I	100	116	CFax
I	117	156	CAddr
I	157	186	CCity
I	187	206	CCount
I	207	216	CZip
I	217	217	CRegion

```
*
* Window . . :
*
* Part . . . :
*
* Event . . :
*
* Description:
*
*****
```

*

C	UPLOAD	BEGACT	PRESS	MAIN	
C	' LIBRARY'	Getatr	'Text'	LIBRARY	
C		If	Library = *Blanks		
C	*MSG0001	Dsply	rc		9 0
C		Else			
c	LC:UC	Xlate	library	library	
C		Eval	tofile = %Trim(library) + '/ QDDSSRC'		
C		Eval	custfile = %Trim(library) + '/ CUSTOMER'		
C		Eval	CMD = 'CRTSRC PF ' + tofile		
C		Exsr	IssCmd		
C		Eval	CMD = 'ADDPFM ' + tofile + ' ' + 'CUSTOMER' +		
C			' SRCTYPE(PF)'		
C		Exsr	ISSCMD		
C		Eval	CMD = 'ADDPFM ' + tofile + ' ' + 'CUST OMLF' +		
C			' SRCTYPE(LF)'		
C		Exsr	ISSCMD		
C		Eval	CMD = 'ADDPFM ' + tofile + ' ' + 'CUSTOML3' +		
C			' SRCTYPE(LF)'		
C		Exsr	ISSCMD		

```

C      'CRTDDS'      Setatr  1          'Checked'
C                      Eval    CMD      = 'OVRDBF QDDSSRC ' + tofile +
C                      MEMBER + ' ' +
C                      'OVRSCOPE(*JOB)'
C                      Exsr     ISSCMD
C
C                      Eval    member = 'CUSTOMER'
C                      Exsr     CPYDDS
C      'CUSTOMER'    Setatr  1          'Checked'
C                      Eval    CMD = 'CRTPF ' + CUSTFILE + ' ' + tofile
C                      Exsr     IssCmd
C      'CRTPF'      Setatr  1          'Checked'
C
C                      Exsr     CpyDta
C                      Eval    member = 'CUSTOMLF'
C                      Exsr     CPYDDS
C      'CUSTOMLF'   Setatr  1          'Checked'
C
C                      Eval    member = 'CUSTOML3'
C                      Exsr     CPYDDS
C
C      'CUSTOML3'   Setatr  1          'Checked'
C
C      'DONE'       Setatr  1          'Checked'
C
C                      EndIf
C
C                      ENDACT
C *****
C
C      * Subroutine . . :
C
C      * Description . . :
C *****
C
C      *INZSR      BEGSR
C
C                      Z-Add    10          SRCSEQ
C
C                      ENDSR
C *****
C
C      * Subroutine . . :
C
C      * Description . . :
C *****

```

```

C      CPYDDS      BEGSR
C      *
C      Eval      CMD      = 'OVRDBF QDDSSRC ' + tofile +
C      MEMBER + ' ' +
C      'OVRSCOPE(*JOB) OPNSCOPE(*JOB)'
C      Exsr      ISSCMD
C      *
C      Eval      PCFile = '.\\' + %Trim(member) + '.DDS'
C      Open      DDSIN
C      Open      QDDSSRC
C      Read      DDSIN
C      Eval      SRCSEQ = 10
C      *
C      Dow      *in99 = *off
C      Write     FMT01
C      Eval      SRCSEQ = SRCSEQ + 10
C      Read      DDSIN
C      EndDo
C      *
C      Close     DDSIN
C      Close     QDDSSRC
C      *
C      ENDSR
*****
*
* Subroutine . . :
*
* Description . . :
*
*****
*
C      ISSCMD      BEGSR
C      *
C      CALL      QCMDDDM
C      PARM
C      PARM      CMD
C      PARM      CMDLEN
C      *
C      ENDSR
*****
*
* Subroutine . . :
*
* Description . . :
*
*****

```

99

99

8 0

```

C      CPYDTA      BEGSR
*
C              Eval      CMD      = 'OVRDBF CUSTOMER ' + custfile +
C                                  ' CUSTOMER OVRSCOPE(*JOB) +
C                                  OPNSCOPE(*JOB)'
C
C              Exsr      ISSCMD
C              Open      Customer
C              Open      PCCust
C              Read      PCCust
*
C                                  99
C
C              DoW      *in99 = *off
C              Write     FMT02
C              Read      PCCust
C              EndDo
*
C              'Data'    Setatr    1          'Checked'
C              Close     Customer
C              Close     PCCust
*
C              Endsr
*****
*
* Window . . . :
*
* Part . . . :
*
* Event . . . :
*
* Description:
*
*
*****
C      CLOSE      BEGACT    PRESS      MAIN
*
C              Move     *on      *inlr
*
C              ENDACT
*****
*
* Window . . . :
*
* Part . . . :
*
* Event . . . :
*
* Description:
*
*
*****
C      MAIN      BEGACT    CREATE      MAIN
*
C      'Library' Setatr    1          'Focus'
*
C              ENDACT

```

Figure 104 (Part 6 of 6). Upload AS/400 Files Program

VRPG Client Lab Setup Instructions for the AS/400 System

Create library GUIDES2 and run the upload program prior to running the setup program.

1. Create libraries A8LABxx for each team.

A team may be one or more students but no more than three people.

2. Create user profiles and job descriptions for each team and specify their corresponding library as their CURLIB.
3. In library GUIDES2, you should find the master objects used for the labs.
4. For each student library, you need to copy the ICF file to their library.

Cleanup

- Delete all student libraries.
- Delete all spooled files from students.
- Delete all user profiles.

CL Source for AS/400 Setup

The following example contains a sample of CL source that can be used to set up the AS/400 system for the labs.

```

5763PW1 V3R1M0 940909                      SEU SOURCE LISTING
SOURCE FILE . . . . . VRPGSETUP/SRC
MEMBER . . . . . VRPGSETUP
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ... +... 7
/*****/
/* Create SETUP program. Pass parameters: */
/* 1-Task: C for Create, D for Delete */
/* 2-Number of users: */
/* 3-Name of AS/400: */
/*****/
PGM PARM(&task &TERMS &MACH)
DCL &task TYPE(*char) LEN(9)
DCL &TERMS TYPE(*dec) LEN(15 5)
DCL &mach TYPE(*char) LEN(8)
DCL &name TYPE(*char) LEN(6)
DCL &lname TYPE(*char) LEN(9)
DCL &num TYPE(*dec) LEN(2 0) VALUE(00)
DCL &cnum TYPE(*char) LEN(2)
/*****/
/* Task is C for Create */
/*****/
IF ((%SUBSTRING(&task 1 1) *EQ 'C') *OR +
    (%SUBSTRING(&task 1 1) *EQ 'c')) THEN(DO)
LOOP: CHGVAR VAR(&num) VALUE(&num + 1)
      CHGVAR VAR(&cnum) VALUE(&num)
      CHGVAR VAR(&name) VALUE('A8LA' *CAT &cnum)
      CHGVAR VAR(&lname) VALUE('A8LAB' *CAT &cnum)
/*****/
/* Create the libraries */
/*****/
CRTLIB LIB(&LNAME) TYPE(*TEST) TEXT(' Library of +
    temporary VRPG user ' *CAT &NAME) +
    AUT(*ALL) CRTAUT(*ALL)
/*****/
/* Create a duplicate object of everything*/
/* in GUIDES2 */
/*****/
CRTDUPOBJ OBJ(*ALL) FROMLIB(GUIDES2) +
    OBJTYPE(*ALL) TOLIB(&lname) DATA(*YES)
/* CRTDUPOBJ OBJ(EVFCICFF00) FROMLIB(QCODE) +
    OBJTYPE(*FILE) TOLIB(&LNAME) +
    NEWOBJ(EVFCICFF) DATA(*YES) */
/* RMOBJ OBJ(&LNAME/EVFCICFF00) OBJTYPE(*FILE) +
    NEWOBJ(EVFCICFF) */
/* */

```

```

/*****
/* Create a job description for each user */
/*****
CRTJOB    JOBD(&LNAME/&LNAME) INLLIBL(&LNAME GUIDE S2)
/*****
/* Create a user profile for each user */
/*****
CRTUSRPRF  USRPRF(&LNAME) USRCLS(*PGMR) CURLIB(&LNAME) +
          TEXT('Temporary user for VRPG CLIENT/2 ') +
          SPCAUT(*JOBCTL) JOBD(&LNAME/&LNAME) +
          GRPPRF(QPGMR) GRPAUT(*ALL)
/*****
/* Add the user to the system distribution*/
/* directory. */
/*****
ADDIRE     USRID(&LNAME &MACH) USRD('For temporary +
          VRPG user ' *CAT &LNAME) USER(&LNAME)
/*****
/* Make the user the owner of all in their*/
/* library */
/*****
CHGOBJOWN  OBJ(QSYS/&LNAME) OBJTYPE(*LIB) +
          NEWOWN(&LNAME) CUROWNOUT(*SAME)
/*****
/* Make the user the owner of all in their*/
/* library */
/*****
GRTOBJAUT  OBJ(&LNAME/*ALL) OBJTYPE(*ALL) USER(*PUB LIC) +
          AUT(*ALL)
          IF (&num *LT &TERMS) THEN(GOTO LOOP)
ENDDO
/*****
/* Task is D for Create */
/* - cleanup libraries & profiles */
/*****
ELSE IF ((%SUBSTRING(&task 1 1) *EQ 'D') *OR +
          (%SUBSTRING(&task 1 1) *EQ 'd')) THEN(DO)
LOOP2:    CHGVAR VAR(&num) VALUE(&num + 1)
          CHGVAR VAR(&cnum) VALUE(&num)
          CHGVAR VAR(&name) VALUE('A8LAB' *CAT &cnum)
          CHGVAR VAR(&lname) VALUE('A8LAB' *CAT &cnum)
          /*
          RMVDIRE USRID(&name &mach) */
          DLTUSRPRF  USRPRF(&LNAME) OWNNOBJOPT(*DLT)
          DLTLIB LIB(&lname)
          IF (&num *LT &TERMS) THEN(GOTO LOOP2)
          ENDDO
ENDPGM

```

Figure 105 (Part 2 of 2). CL Setup Source

CL Source for AS/400 Setup Command

The following example contains a sample of CL source that can be used to create an AS/400 setup command.

```

CMD      PROMPT(' CREATE VRPG LAB ENVIRONMENT')
PARM     KWD(TASK) TYPE(*CHAR) LEN(1) RSTD(*YES) +
          DFT('C') VALUES('C' 'c' 'D' 'd') +
          PROMPT(' TASK C OR D')
PARM     KWD(USERS) TYPE(*DEC)  +
          DFT(10) +
          PROMPT(' NUMBER OF USERS')
PARM     KWD(MACH) TYPE(*CHAR) LEN(10) +
          DFT(A) +
          PROMPT(' MACHINE NAME')

```

Figure 106. CL Setup Command Source

Part 3. Information for Service Providers

This part of the Redbook is intended to be a guide for service providers on assembling a service offering. The service offering can include installation of ADTS/CS as well as education to help build skills for exploiting the different tools in ADTS/CS.

Chapter 14. Service Offering

This document can be used by a service provider to assemble a service offering on the installation and setup of VRPG Client and education on creating VRPG Client applications.

This document does not give guidance on how to price a service offering. Sizing the job depends on many factors including the skill and experience level of the service provider and the characteristics of the customer's Information Technology (IT) and business environment. It is up to you to assess these variables and estimate the size and cost of the job.

VRPG Client Installation and Configuration Service Offering

There are certain phases to go through to do an installation and configuration service offering.

- Phase 1 - IT environment inventory
- Phase 2 - Hardware and software acquisition
- Phase 3 - Installation and configuration

After the IT environment inventory, determine what if any additional hardware or software is required and who has the responsibility to acquire it and attach it to the system. Once the hardware and software are in place, determine which installation and configuration scenario to use and follow the appropriate steps. Included on the diskette is a sample program to verify whether the installation and configuration have been done successfully.

VRPG Client Education Service Offering

The exercises included in Chapter 13, "VRPG Client/2 Labs" can be used to teach the customers and solution providers how to use the product or as a self-study tutorial. Freelance presentations are included on the diskette that are to be used as introductions to the lab exercises.

Inventory for Installation and Configuration

The first phase in the installation and configuration offering is to understand the customer environment. Do an inventory of their current host and client machines that they plan to use for VRPG Client development. Inventory their hardware, software, middleware, and products currently on those machines.

- Does their current host and client machines meet the requirements for VRPG development?

This includes the number of machines, memory, DASD, and so on.

- What additional products do they plan to have on those machines?
Have them project out to the next year.

- Do they plan to use SQL?
- Do they plan to access data on the client as well as the host?
- What additional communications needs do they have to the host?

- Do they plan to use a LAN server or Integrated PC File Server for the installation or load the product on each machine?

This determines how updates to the VRPG Client product are done.

- What type of change management is needed/planned?
- What are their performance expectations for this development environment?
- Do they have Local Area Network (LAN) experience?
- What communication protocols do they want to use?
- How many developers are on the system at one time?
- How many compiles are done at the same time?
- What other work besides VRPG Client development is going on at the same time?

The goal of the IT environment inventory is to produce a gross estimate of the resources that are required to make the customer's environment suitable for easy development and maintenance of VRPG Client applications. From the inventory analysis, you produce an IT environment overview report that is used to do this estimate (the initial proposal). The initial proposal gives the customer a rough idea of the cost of the setting up an environment for doing VRPG Client development. This allows the customer or solution provider to decide whether to proceed with the installation or look at other alternatives. At the end of this chapter, there is a discussion of the contracts that you need to put in place as the project develops. For now, just be aware that you should put a contract with a statement of work in place for the IT environment inventory phase of the project. This allows you to bill the customer for your services, even if the decision is made not to go ahead with the installation.

The most important task in the IT environment inventory phase is to gather the necessary information to understand the customer's environment. Much of the information must be gathered manually by having the customer complete a survey of their environment; and some of the information emerges as a part of informal discussions. You need to gather enough detail from the customer to feel comfortable that you understand their environment.

You need the following information to help you produce a high-level estimate of the resources that are required to complete the installation.

Company Description

This information should be collected using some type of survey.

- Nature of the business
- Key resources (management, technical)
- Processing cycle (for example, daily, monthly, yearly)
- Availability requirements
- Performance requirements
- Urgency of installation and first project
- Language spoken
- Size of DP staff
- Skills

AS/400 Systems Configuration (for Each System)

This information can be gathered using the CL commands that follow:

- CPU specifications (indicate if upgrades are planned)
- Model
- Memory
- DASD (size and utilization)
- IOPs
- CPU utilization
- Security level

```
DSPHDWRSC TYPE(*AHW) OUTPUT(*PRINT)
DSPSYSVAL QSECURITY OUTPUT(*PRINT)
```

Network Configuration

This information can also be gathered by CL commands referenced in the system configuration section.

- AS/400 systems
- OEM hardware

Type of Service to Be Performed

- Inventory analysis
- Hardware software acquisition
- Installation and configuration
- Verification testing

Some customers may want you to do an estimate of the size of the entire project during the IT environment inventory phase. Others may want you to size the cost one step at a time. For each phase, it is important to understand the resources that the customer has available.

System Software (for Each System)

This information can be gathered using the CL commands that follow:

- OS/400 Release (indicate if upgrades are planned)
- PTF level
- S/3X environment
- License program products
- Primary national language
- Secondary national language
- Tools

```
WRKSYSACT OUTPUT(*FILE) INTERVAL(60) NBRITV(60)
DSPSFWRSC OUTPUT(*PRINT)
DSPPTF OUTPUT(*PRINT)
```

IT Environment Overview Report

This report is delivered to the customer to document your understanding of the data you have collected. The customer reviews and revises this document as appropriate to produce an accurate representation of the environment and needs.

Initial Proposal

When the IT environment overview report is complete and the customer agrees it is accurate, you can generate an initial proposal. This proposal gives a high-level estimate of the resources that are required to complete the work and should be broken down by (inventory analysis, hardware and software acquisition, installation and configuration, and education). The customer may want you to estimate one phase at a time rather than the entire project. You should be sure you understand what services the customer wants your estimate to include.

The estimate should be stated in terms of person days and or months and should indicate a range rather than exact number. The purpose of this estimate is to provide the customer with enough information to decide whether to proceed with an installation and configuration or search for another offering that meets the needs of their business. In addition to installation and configuration services and education, the service provider is encouraged to perform the service (for fee) of locating alternate solutions for customers who decide the installation and configuration or education is too expensive.

Sizing of Installation and Configuration Offering

The IT environment overview report can be used to determine how much effort and time is involved in the installation and configuration.

Other Costs

Other costs that need to be considered are subjective. Additional or upgraded hardware may be required, both during the installation or afterwards if the customer's DASD or memory requirements grow as a result of additional development work. There may be schedule considerations (that is, work must be completed by a particular date), or where the work is performed may be a critical factor. Other issues such as performance or availability requirements must be understood. You may also need to consider additional training costs.

Contracts

It is necessary for you to develop a separate contract with a statement of work for each of the phases of the installation and configuration and education if your customer wants to base a decision to proceed with the next phase of the project on the results of the previous stage. For instance, you need to put a contract in place for the work you do in the IT environment inventory phase so you are able to charge the customer for the work that is involved in producing the initial proposal. The customer can decide whether to proceed with the next phase (the installation and configuration) based on the results of the initial proposal. Likewise, you need a contract in place for the education, and the customer can decide whether to proceed with the education based on the results of the

installation and configuration verification. Finally, a contract needs to be developed for any follow-on education or support activities. These may be done separately or combined into one contract. In all phases, the statement of work is an important communications vehicle and is key to developing an effective working relationship between the service provider and the customer. It not only describes the deliverables of the project, but defines the way the teams divide the work and share information. The following is an outline of the contents of a contract and a statement of work.

Contract Content

- Purpose
- Work scope and structure
- Rights/intellectual property
- Indemnification
- Warranty
- Term and termination
- Contacts
- Name of contract coordinator
- Limitations
- Freedom of action
- Trademarks
- Expenses
- Assignment
- Governing law
- Jury trial
- Subsidiary
- Entire agreement
- Signatures

Statement of Work

The statement of work contains the details of the arrangement between you and the customer. The following example contains the information that may be included in a statement of work.

- Technical coordinators (list names and addresses)
- Tasks:
 - Outline the tasks that are completed, the criteria for acceptance, and the completion dates. Be sure to break the project down into milestones that can be tracked to make sure the schedule is maintained.
- Organization, responsibility, and resources: Resource requirements should be established and the division of responsibilities documented. Skill requirements must also be outlined and training plans established if needed. Leadership responsibilities and the reporting structure should be well understood. Be sure to devote adequate resources for project management to keep the project on schedule and to maintain the relationship between you and the customer.

The customer's development environment, including change control tools and processes, must be well understood when assessing the level of effort in a project.

Hardware resource requirements must be identified. Where is the service performed? Who is providing the hardware? Where is it located? Space,

facilities and support should all be planned for. Travel expenses should be budgeted as well.

- Project management system:

The customer and the service provider must establish how project milestones are tracked and communicated. Processes for the management of changes, problems, and issues should be put in place. Being well organized in this area is key to maintaining a positive relationship between the your company and the customer. You may find your project management styles are quite different, so it is important to agree from the start on how the project is run.

- Risks, assumptions, and dependencies:

The statement of work must include the assumptions upon which the project and its plans are based. External dependencies and risk assessments should also be part of the document.

- Payment:

Document the amount and schedule for payment.

- Intellectual property:

You have to be aware of intellectual property concerns. If your employees are exposed to the customer's source code, they may be restricted in what they can work on in the future because they have been "contaminated". Also, be sure your customer has the right to modify the code if it was purchased from a third party.

The statement of work should be written so that modifications and additions are possible as the project proceeds.

Chapter 15. Post Services

This chapter describes the support you may provide to the customer once the installation, configuration, and initial education is complete. You need to be clear on the scope and duration of this service. Some customers want it for a short duration, while others may want you to provide continuing support. It is important that you clearly state your intentions in the statement of work so that it is easy to identify the end of the project and, hence, the end of your relationship with the customer.

If the post services are not a part of the initial proposal, you may need to submit a new financial proposal to the customer spanning the explicit services that the service provider wants to provide.

Further Education or Support

If you choose to offer additional education or support to the customer, it is advisable to outline the provisions in a separate contract.

The following list contains some of the elements of a support service offering:

- The process through which the help support questions/problems are evaluated including:
 - Design forms for formal submission and documentation of questions or problems.
 - Define a set of rules to evaluate problems.
 - Determine if it is an operational problem by checking the operator input.
 - If the problem is operational, evaluate the job log.
 - Look at the source code as well as the call stack.
 - Understand and re-create the events that caused the problem to occur.
 - Ensure that the problem is not corrupting data as quickly as possible.
- Estimate and allocate resources.
- Include a statement of limitation of scope and liability in the contract. Be sure you have a way for charging for work that is outside the scope of the contract or work that is caused by misinformation from the customer.
- The duration of the support service offering depends upon the following factors:
 - Size of the application software
 - Lines of code provided
 - Relationship with your customer
 - Your own preferences in providing such a service
- Resource requirements projected
- Financial proposal
- Exact description of start and end dates of the service

During the help support period, the service provider may recommend that the customer enhance the application. You may give the customer some kind of a rough estimate of the resource required and put a separate contract in place for this work. This is a good time to point out current trends in the industry and possible enhancements to the current application or other applications.

Additional Service Offerings Out of Post Relationship

Since you may be in a long term relationship with the customer, the customer might want you to perform other services such as:

- Further installations and configurations:
As the customer acquires additional PCs, can they do the installation and configuration or do they want you to do the work?
- Establish proper backup procedures.
- Standardize the applications.
- Functionally enhance the application software.
- Performance tuning.
- Capacity planning.
- Availability enhancements.
- Create a comprehensive testing methodology.
- Find another vendor that can help the customer with problems, or even search for an alternate solution.

There are many types of services that you do not choose to perform. It is good to be aware of other service providers that specialize in these areas so that you can refer the customer to them.

Appendix A. Special Notices

This publication is intended to help service providers successfully set up a complete development environment for the Application Development ToolSet Client/Server (ADTS CS) for their customers. The material is also useful for in-house DP Professionals who are installing ADTS CS on their own and have the necessary knowledge, skills, and experience with OS/400, OS/2, Client Access/400, CODE/400, and VRPG Client/2. The information in this publication is not intended as the specification of any programming interfaces that are provided by OS/400, OS/2, Client Access/400, CODE/400, and VRPG Client/2. See the PUBLICATIONS section of the IBM Programming Announcement for the products mentioned above for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AS/400	Client Access
DB2	DB2/400
IBM	OS/2
OS/400	SAA
VisualAge	WIN-OS/2

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Lotus, Freelance	Lotus Development Corporation
------------------	-------------------------------

Other trademarks are trademarks of their respective companies.

Appendix B. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

International Technical Support Organization Publications

For information on ordering these ITSO publications see "How to Get ITSO Redbooks" on page 235.

- *Inside Client Access/400 for Windows 3.1*, SG24-4429-00
- *Personal Communications for OS/2 and CM/2: Product Differentiation Guide*, GG24-2546-00
- *Client Access/400 API Planning Guide*, GG24-4422-00
- *Using V2R3 DOS and OS/2 PC Support/400 under OS/2 2.1, including Communications Manager/400, Communications Manager/2, and RUMBA/400 for OS/2*, GG24-4070-01

Online books supplied with ADTS CS

This section contains an extensive list of softcopy and hardcopy documents which may prove useful for further reading.

ADTS CS

- Install Document (Install.Doc)
- ADTS CS "Start Here" (EVFSTR40.inf)

WorkFrame

- WorkFrame "How Do I...?" (IWFHDI.inf)
- CODE and WorkFrame (EVFWF400.inf)

CODE/400

- CODE Read-Me-First (EVFREAD.inf)
- CODE and WorkFrame (EVFWF400.inf)
- CODE Tips & Techniques (EVFFAQ.inf)
- CODE and VRPG EXTRAS function (EVFEXTRA.inf)
- CODE Editor "How Do I...?" (EVFELHDI.inf)
- CODE Editor Reference (EVFELREF.inf)
- CODE Command-Line Invocation (EVFGS400.inf)

LPEX Editor

- LPEX "How Do I...?" (LPEXHDI.inf)
- LPEX Reference (LPXCREF.inf)
- CODE Editor "How Do I...?" (EVFELHDI.inf)
- CODE Editor Reference (EVFELREF.inf)
- CODE and VRPG EXTRAS function (EVFEXTRA.inf)

VRPG

- VRPG Read-Me-First (README.inf)
- VRPG Getting Started (SC092195.inf)
- VRPG Visual Programming (SC092131.inf)
- VRPG Tutorial (FVDETUT.inf)
- VRPG Parts Reference (SC091846.inf)
- VRPG Language Reference (SC091847.inf)
- VRPG Parts Development NewsLetter (TOOLKIT.inf)
- VRPG Messages (FVDMSG.inf)
- CODE and VRPG EXTRAS function (EVFEXTRA.inf)

IPF (Information Presentation Facility)

- IPF (IPFC20.inf)

Other Useful Publications

ADTS CS

- *Introduction to ADTS CS for OS/400*, GC09-2189-00
- *Getting Started with VRPG Client for OS/2*, SC09-2195-00
- *VRPG Client for OS/2 Language Reference*, SC09-1847-00
- *CODE/400 Debug Tool*, SC09-1905-00
- *CODE/400 Self Study Guide*, SC09-1911-00
- *Client/Server Visual Programming with VRPG Client for OS/2*, SC09-2131-00
- *Installing ADTS CS for OS/400*, SC09-2188-00

AS/400 System

Well, a good starting point is:

- *AS/400 Publications Reference*, SC41-3003

You can find this on the Internet at IBM InfoLink:

1. Go <http://www.ibmmlink.ibm.com>.
2. Select an appropriate region or the U.S.
3. Select *IBM Manuals*.
4. From the *Library Catalog*, select **AS/400**.
5. Find and select document **qbkaie00**.

OS/2 Warp

The OS/2 Warp technical manuals are available to the public on the Internet at IBM InfoLink.

Follow the preceding instructions for the AS/400 system, but select **PWS** instead of **AS/400**.

CM/2: Softcopy Provided

- Command Reference (CMCMDREF.INF)
- Glossary (GLOSSARY.INF)
- Message Reference (CMMSGREF.INF)
- Overview (OVERVIEW.INF)
- Supported Modems (WNMODEM.INF)

- What's New (WHATSNEW.INF)

Communications Server

<http://www.raleigh.ibm.com/cm2/cm2pubs.html>

This site provides the facility to download most relevant books, usually in BookManager (.BOO) format, as well as some 'View Now'.

Personal Communications

<http://www.raleigh.ibm.com/pcf/pcfpubs.html> has downloads (BookManager and PostScript) plus some 'View Now'.

Client Access/400

CA/400 Family

- *LPS: Client Access/400 Family V3R1*, GC41-3052-02
- *LPS: Client Access/400 ToolKit V3R1*, GC41-3054-00
- *LPS: Client Access/400 Ultimedia Tools V3R1*, GC41-3075-00
- *LPS: Client Access for OS/400 V3R6*, GC41-4052-00
- *LPS: Client Access Ultimedia Tools for OS/400 V3R6*, GC41-4075-00
- *Client Access/400 Console*, G325-6200-01
- *Client Access for OS/400 GUI Features:Positioning Paper*, G325-6308-00
- *Client Access/400 PC5250 User's Guide*, SC41-3552-00
- *Client Access/400 PC5250 Reference Guide*, SC41-3553-01
- *Client Access/400 PC5250 Setup Guide*, SC41-3555-00
- *Client Access/400 Family of Products*, SC41-3560-00

CA/400 for OS/2

- *LPS: Client Access for OS/2*, GC41-4052-00
- *Client Access/400 Optimized for OS/2 - Getting Started*, SC41-3510-01
- *Client Access/400 Optimized for OS/2 API and Technical Reference*, SC41-3511-01
- *Client Access/400 for OS/2 Setup V3R1*, SC41-3520-01
- *Client Access/400 for OS/2 User Guide V3R1*, SC41-3521-01
- *Client Access/400 for OS/2 Setup (DBCS) V3R1*, SC41-3522-01
- *Client Access/400 for OS/2 User Guide (DBCS) V3R1*, SC41-3523-01
- *Client Access/400 for DOS and OS/2 API Reference V3R1*, SC41-3562-00
- *Client Access/400 for DOS and OS/2 Technical Reference V3R1*, SC41-3563-01

CA/400 for Windows

- *Client Access/400 for Windows 3.1 User Guide*, SC41-3532-00
- *Client Access Windows 3.1 Client for OS/400 ODBC User's Guide*, SC41-3533-00
- *Client Access/400 for Windows 3.1 - Getting Started*, SC41-3530-01
- *Client Access/400 for Windows 3.1 API and Technical Reference.*, SC41-3531-01
- *Client Access/400 for Windows 3.1: TCP/IP Setup*, SC41-3580-00

PC Support/400

General

- *AS/400 PC Support/400 API Reference*, SC41-8254-02

PCS for OS/2

- *AS/400 PC Support/400 OS/2 Installation and Administration Guide*, SC41-0007-02
- *AS/400 PC Support/400 OS/2 Installation and Administration Guide (PS/55)*, SC41-0009-02
- *AS/400 PC Support/400 User's Guide for OS/2 (PS/55)*, SC41-2415-02
- *AS/400 PC Support/400 Technical Reference for DOS and OS/2*, SC41-8091-02
- *AS/400 PC Support/400 User's Guide for OS/2*, SC41-8200-02

ADM

- *Application Development Manager/400 User's Guide*, SC09-2133-00

Internet Resources

The following entries are provided for those users who have internet access. In general, internet access will provide the user with the current versions of manuals and often there will be recent articles which appear prior to hardcopy publication.

VRPG Home Page

This is a very important link! Put in in your browser's hotlinks:

<http://www.software.ibm.com/ad/varpg/>

IBM Public Information and Services Main Menu

A good place to begin for General Information is at the IBMLink WorldWide Information page at:

<http://www.ibmlink.ibm.com>

Select your region or use the main U.S. link at the bottom of the list.

You will then see *IBM Public Information and Services Main Menu*.

This links to many excellent places, including (at the time of writing):

- IBM InfoLink Library: Catalog of Online Viewable Documents
- AS/400 Publications (there are many!)
- PWS (OS/2, LAN,MMPM/2) publications
- ITSO RedBooks

Product Descriptions

See IBM Software at <http://www.software.hosting.ibm.com/ad/as400> and IBM Direct at <http://direct.boulder.ibm.com/us/devsol/integrate/>

Education

IBM Course G4501 OS/2 Communications Manager Workshop

<http://is.ibm.com.au/ProdServe/educat/G4501.html>

IBM Course AS/400 V3R6 Application Development using VRPG

Service

AS/400 Support Line (Rochester)

<http://as400service.rochester.ibm.com>

IBM Direct: Customer Service

<http://direct.boulder.ibm.com/us/customer>

Technical Support Databases [IBM Raleigh]

<http://www.raleigh.ibm.com/cm2/cm2serv.html>

This page also lists the sources of updates and fixes for desktop products including OS/2 Warp and the communications products.

General

IBM Link *<http://www.ibm.link.ibm.com>*

AS/400 Partners in Development

<http://www.softmall.ibm.com/As400/>

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL <http://www.redbooks.ibm.com>.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

To get LIST3820s of redbooks, type one of the following commands:

```
TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
```

To get lists of redbooks:

```
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
```

To register for information on workshops, residencies, and redbooks:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
```

For a list of product area specialists in the ITSO:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
```

- **Redbooks Home Page on the World Wide Web**
<http://w3.itso.ibm.com/redbooks>
- **IBM Direct Publications Catalog on the World Wide Web**
<http://www.elink.ibm.link.ibm.com/pb1/pb1>
IBM employees may obtain LIST3820s of redbooks from this page.
- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibm.link.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	Internet
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications Publications Customer Support P.O. Box 29570 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark
--	--	--

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
(+45) 48 14 2207 (long distance charge)	Outside North America

- **1-800-IBM-4FAX (United States)** or **(+1)001-408-256-5422 (Outside USA)** — ask for:

Index # 4421 Abstracts of new redbooks
Index # 4422 IBM redbooks
Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com

- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibm.link.ibm.com/pbl/pbl

- **Internet Listserver**

With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibm.link.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name	Last name
------------	-----------

Company

Address

City	Postal code	Country
------	-------------	---------

Telephone number	Telefax number	VAT number
------------------	----------------	------------

- Invoice to customer number

- Credit card number

Credit card expiration date	Card issued to	Signature
-----------------------------	----------------	-----------

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

Glossary

This glossary includes terms and definitions from:

- The *American National Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York, 10018. Definitions are defined by the symbol (A) after the definition.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Committee (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition indicating that the final agreement has not yet been reached among participating National Bodies of SC1.
- *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.
- *Object-Oriented Interface Design IBM Common User Interface Guidelines*, SC34-4399-00, Carmel, IN: Que Corporation, 1992.
- *Communications Manager/2 Glossary*, SC31-6155-02, IBM Corp.

If the definitions in this glossary differ from those in the &bigsys. &b4006., &b4006n., use the ones in this glossary.

A

action. (1) Synonym for *action subroutine*. (2) An executable program or command file used to manipulate a project's parts or participate in a build.

action subroutine. Logic that you write to respond to a specific event.

active window. The window with which a user is currently interacting. This is the window that receives keyboard input.

Adapter. A circuit card that connects your workstation to a network.

Adapter Number. For token-ring or other LAN types, Ethernet (ETHERAND) network, and PC Network, the adapter number can be either 00 or 01. For SDLC or

ISDN data link control (IDLC), the adapter number can be 00 through 127. Cannot be specified for an X.25 or a twinaxial connection

Address Format. The format of the LAN destination address at the remote workstation. The two address formats are: (1) Token-ring. The token-ring address format is most significant bit. This means that the highest bit within each byte of the address is sent across the LAN first when establishing the connection. (2) Ethernet. The Ethernet address format is least significant bit. This means that the lowest bit within each byte of the address is sent across the LAN first when establishing the connection.

Adjacent Node ID. The eight hex characters used by the partner node in the exchange identification (XID) for activating a link. In station negotiations, the adjacent node ID determines which link workstation is primary and which is secondary. The workstation with the higher value becomes the primary link workstation. If equal, both workstations generate a random number to determine which is primary.

anchor. Any part that you use as a reference point for aligning, sizing, and spacing other parts.

application. A collection of software components used to perform specific user tasks on a computer.

ASCII (American National Standard Code for Information Interchange). The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

audio part. A part that gives a program the ability to process wave files (.WAV) and MIDI files (.MID).

B

.BMP. The file extension of a bitmap file.

build. The process by which the various pieces of source code that make up components of a VRPG Client application are compiled and linked to produce an executable version of the application.

button. (1) A mechanism on a pointing device, such as a mouse, used to request or start an action. (2) A graphical mechanism in a window that, when selected, results in an action. An example of a button is an OK push button that, when selected, initiates an action.

C

canvas part. A part onto which you can drag-and-drop various other parts, position them, and organize them to produce a graphical user interface. A canvas part occupies the client area of either a window part or a notebook page part. See also *notebook page with canvas part* and *window with canvas part*.

cascaded menu. A menu that appears from, and contains choices related to, a cascading choice in another menu. Cascaded menus are used to reduce the length of a pull-down menu or a pop-up menu.

check box part. A square box with associated text that represents a choice. When a user selects a choice, an indicator appears in the check box to indicate that the choice is selected. The user can clear the check box by selecting the choice again. In VRPG Client, you drag a check box part from the parts palette or parts catalog and drop it onto a design window.

click. To press and release a mouse button without moving the pointer off of the choice or object. See also *double-click*.

client. (1) A system that is dependent on a server to provide it with data. (2) The PWS on which the VRPG Client and VRPG Client applications run. See also *DDE client*.

client area. The portion of the window that is the user's workspace, where a user types information, and selects choices from selection fields. In primary windows, the area where an application programmer presents the objects that a user works on.

client/server. The model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server. See also *client*, *server*, *DDE client*, *DDE server*.

clipboard. An area of storage provided by the system to hold data temporarily. Data in the clipboard is available to other applications.

cold-link conversation. In DDE, an explicit request made from a client program to a server program. The server program responds to the request. Contrast with *hot-link conversation*.

color palette. A set of colors that can be used to change the color of any part in your application's GUI.

combination box. A control that combines the functions of an entry field and a list box. A combination box contains a list of objects that a user can scroll through and select from to complete the

entry field. Alternatively, a user can type text directly into the entry field. In VRPG Client, you can drag a combination box part from the parts palette or parts catalog and drop it onto a design window.

Common User Access architecture (CUA architecture). Guidelines for the dialog between a human and a workstation or terminal.

compile. To translate a source program into an executable program (an object program).

component. A functional grouping of related files within a project.

component reference part. A part that enables one component to communicate with another component in a VRPG Client application.

CONFIG.SYS. The configuration file located in the root directory of the boot drive for the DOS and OS/2 operating systems. It contains information required to install and run hardware and software.

configuration. The manner in which the hardware and software of an information processing system are organized and interconnected.

container. A part that stores related records and displays them in a details icon or tree view.

CUA architecture. Common User Access architecture.

Connection Manager. A service of CM/2 that establishes connections for applications over ISDN lines.

Connection Network Name. Two parts: (1) network ID (can be the network ID from the Local Node Characteristics window) and (2) a name identifying the LAN you are connected to. This can be the *C&SM LAN ID* from the LAN DLC adapter profile.

Conversation Security Verification. Indicates the type and amount of conversation security information that your workstation requires when the partner LU requests a communication session. When this option is selected, a conversation request from this partner LU can include either a password or an indication that the password has been verified. Otherwise, a conversation request from this partner LU must include a password. Previous verification is not accepted by your workstation. A password must be provided each time.

Note: When a conversation request contains both user ID and password, the system checks both fields regardless of which setting you choose.

CPI-C. Common Programming Interface Communications. Protocols and conventions to communicate with other network stations. CPI-C is the API provided with APPN. You can configure CPI-C

for all workstation connection types except asynchronous and DFT.

CPI Side Information. Includes the LU names and TP names of programs that you can access through the CPI Communications subsystem. Provides the CPI Communications subsystem with the information it needs to establish a conversation with a partner program. The information that is provided includes the name of the partner program and the name of the LU supporting the session used by the partner program.

cursor. The visible indication of the position where user interaction with the keyboard is shown.

C&SM LAN ID. In CM/2, this identifies the LAN that the adapter belongs to. C&SM LAN ID is common among all adapters using a given LAN. The LAN ID is used to identify the LAN in network management alerts.

D

database. (1) A collection of data with a given structure for accepting, storing, and providing, on demand, data for multiple users. (2) All the data files stored in the system.

data object. An object that conveys information, such as text, graphics, audio, or video.

DBCS. Double-byte character set.

DDE. Dynamic data exchange.

DDE client. An application that initiates a DDE conversation. Contrast with *DDE server*. See also *DDE client part*, *DDE conversation*.

DDE client part. A part used to exchange data with other applications, such as spreadsheet applications, that support the dynamic data exchange (DDE) protocol.

DDE conversation. The exchange of data between a DDE client and a DDE server. See also *cold-link conversation* and *hot-link conversation*.

DDE server. An application that provides data to another DDE-enabled application. Contrast with *DDE client*. See also *DDE conversation*.

default. A value that is automatically supplied or assumed by the system or program when no value is specified by the user. The default value can be assigned to a push button or graphic push button.

default action. An action that is performed when some action is taken, such as pressing the Enter key.

dereferencing. The action of removing the association between a part and an AS/400 database field.

design window. The window in the GUI designer on which parts are placed to create a user interface.

details view. A standard contents view in which a small icon is combined with text to provide descriptive information about an object.

dimmed. Pertaining to the reduced contrast indicating that a part cannot be selected or directly manipulated by the user.

direct manipulation. The use of techniques that allow a user to work with an object by dragging it with a mouse or interacting with its pop-up menu.

DLL. Dynamic link library.

double-byte character set (DBCS). A set of characters in which each character is represented by two bytes. Languages such as Japanese, Chinese, and Korean that contain more symbols than can be represented by 256 code points require double-byte character sets. Because each character requires two bytes, typing, displaying, and printing of DBCS characters requires hardware and programs that support DBCS. Four double-byte character sets are supported by the system: Japanese, Korean, Simplified Chinese, and Traditional Chinese. Contrast with *single-byte character set (SBCS)*.

double-click. To quickly press a mouse button twice.

drag. To use a mouse to move or to copy an object. For example, a user can drag a window border to make it larger by holding a button while moving the mouse. See also *drag and drop*.

drag and drop. To directly manipulate an object by moving it and placing it somewhere else using a mouse.

drop-down combination box. A variation of a combination box in which a list box is hidden until a user takes explicit acts to make it visible.

drop-down list. A single selection field in which only the current choice is visible. Other choices are hidden until the user explicitly acts to display the list box that contains the other choices.

dynamic data exchange (DDE). The exchange of data between programs or between a program and a data file object. Any change made to information in one program or session is applied to the identical data created by the other program. See also *DDE conversation*, *DDE client*, *DDE server*.

Dynamic link library (DLL). A file containing executable code and data bound to a program at load

time or run time rather than during linking. The code and data in a dynamic link library can be shared by several applications simultaneously.

E

EBCDIC. Extended binary-coded decimal interchange code. A coded character set of 256 8-bit characters.

emphasis. Highlighting, color change, or other visible indication of conditions relative to an object or choice that affects a user's ability to interact with that object or choice. Emphasis can also give a user additional information about the state of a choice or an object.

entry field part. An area on a display where a user can enter information, unless the field is read-only. The boundaries of an entry field are usually indicated. In VRPG Client, you drag an entry field part from the parts palette or parts catalog and drop it onto a design window.

event. A signal generated as a result of a change to the state of a part. For example, pressing a button generates a *Press* event.

exception. (1) In programming languages, an abnormal situation that may arise during execution, that may cause a deviation from the normal execution sequence, and for which facilities exist in a programming language to define, raise, recognize, ignore, and handle it. (2) In VRPG Client, an event or situation that prevents, or could prevent, an action requested by a user from being completed in a manner that the user would expect. Exceptions occur when a product is unable to interpret a user's input.

.EXE. The extension of an executable file.

export. A function that converts an internal file to some standard file format for use outside of an application. Contrast with *import*.

F

field. (1) An identifiable area in a window, such as an entry field where a user types text. (2) A group of related bytes, such as a name or amount, that is treated as a unit in a record.

file. A collection of related data that is stored and retrieved by an assigned name. A file can include information that starts a program (program file object), contains text or graphics (data-file object), or processes a series of commands (batch file).

focus. Synonym for *input focus*.

font palette. A set of fonts that can be used to change the font of a part in your application's GUI.

G

graphical user interface (GUI). A type of user interface that takes advantage of high-resolution graphics. A graphical user interface includes a combination of graphics, the object-action paradigm, the use of pointing devices, menu bars and other menus, overlapping windows, and icons.

graphic push button part. A push button labeled with a graphic that represents an action that is initiated when a user selects it. Contrast with *push button part*.

group box part. A rectangular frame around a group of controls to indicate that they are related and to provide an optional label for the group. In VRPG Client, you drag a group box part from the parts palette or parts catalog and drop it onto a design window.

group marker. A mark that identifies a part as being the first one in a group. When a user moves the cursor through a group of parts and reaches the last part, the cursor returns to the first part in the group.

GUI designer. A suite of tools used to create interfaces by dragging-and-dropping parts from the parts palette to the design window.

H

hide button. A button on a title bar that a user clicks on to remove a window from the workplace without closing the window. When the window is hidden, the state of the window, as represented in the window list, changes. Contrast with *maximize button* and *minimize button*.

hot-link conversation. In DDE, an automatic update of a client program by a server program when data changes on the server. Contrast with *cold-link conversation*.

I

.ICO. The file extension of an icon file.

icon. A graphical representation of an object consisting of an image, image background, and a label.

icon view. A standard contents view in which each object contained in a container is displayed as an icon.

image part. A part used to display a picture from a .BMP or .ICO file on a window.

import. A function that converts AS/400 display file objects to the appropriate VRPG Client part. Contrast with *export*.

inactive window. A window that cannot receive keyboard input at a given moment.

index. The identifier of an entry in VRPG Client parts such as list boxes or combination boxes.

information area. A part of a window in which information about the object or choice that the cursor is on is displayed. The information area can also contain a message about the normal completion of a process. See also *status area*.

Information Presentation Facility (IPF). A tool used to create online help on a programmable workstation.

Information Presentation Facility (IPF) file. A file in which the application's help source is stored.

.INI. The file extension for a file in the OS/2 operating system containing application-specific information that needs to be preserved from one call of an application to another.

input focus. The area of a window where user interaction is possible from either the keyboard or the mouse.

input/output (I/O). Data provided to the computer or data resulting from computer processing.

IPF. Information Presentation Facility

item. In dynamic data exchange, a unit of data. For example, the top left cell position in a spreadsheet is row 1, column 1. This cell position may be referred to as item R1C1.

L

link event. An event that a target part receives whenever the state of a source part changes.

list box part. A control that contains scrollable choices that a user can select. In VRPG Client, you can drag the list box part from the parts palette or parts catalog and drop it onto a design window.

LAN destination address. Twelve-character hexadecimal address of the adapter on the node (communications controller or gateway) to which you are defining a connection. For a 3270 emulation configuration, the LAN destination address is the address of the network adapter for your SNA gateway or your SNA controller. For a 5250 emulation configuration, the LAN destination address is the address of the adapter on the AS/400 system you connect to, or it is the network node you connect through to an AS/400 system.

The first character must be a digit between 0 and 7. Depending on your network, the LAN destination address is one of the following: (1) The node address or the MAC address defined in the configuration for the controller's network adapter on an IBM Token-Ring network or an IBM PC network or (2) A permanent address that is encoded in the controller's network adapter by the manufacturer (a 'burned-in' address) or (3) An address assigned within your network to the controller's network adapter (a 'locally-administered' address). In CM/2, you give either the Adjacent node ID field or the LAN destination address field.

LAPS. LAN Adapter and Protocol Support. Part of NTS/2. For OS/2 Warp Connect and *Merlin*, this is replaced by MPTN.

LEN node. Low Entry Networking node: can participate in APPC communications, but cannot make direct use of the APPN services provided by a network node server. An example of a LEN node is an OS/2 EE 1.2 or 1.3 Communications Manager node without Networking Services/2 (NTS/2) installed.

Link Name. User-defined name identifying the connection you are defining.

Local LUs. Fully-qualified names of LUs that reside on your workstation. The primary LU for your workstation is its control point (CP). The CP assists with initiating and ending your LU sessions. In most cases, the CP LU provides all the communications functions a workstation needs. Some workstations need additional LUs for the following reasons: (1) Your workstation is using more than one session at a time with a specific partner LU, (2) The software applications on your system require additional LUs.

Local Node Alias. An optional alternative name or alias name by which local programs can refer to your node or control point LU. Alias names are generally used for convenience or easy recognition within a network. If this field is left blank, **Local node name** is used as the Local node alias.

Local Node ID. Specifies the last eight characters used in the exchange identification (XID) for activating a link. It must be a hexadecimal value between 00000000 and FFFFFFFF. The first three characters default to X'05D'. In link station negotiations, these eight characters determine which link workstation is primary and which is secondary. The workstation with the higher value becomes the primary link workstation.

Local LU name. Note that the default Local LU Name is obtained from the **Local Node Name**.

Local Node Name. The name that other nodes in your network use to address your node. Also called your control point (CP) name. CM/2 uses this as the default local LU name. Must be unique.

Note: If you are configuring management support for NetWare, the local node name is used as the user ID for creating a connection to a NetWare server. The user ID and its related password must be configured on the NetWare server.

Local sap. Hex address of the Service Access Point (hex) that a remote workstation must use to communicate with your workstation. Valid values range from X'04' to X'9C' and must be divisible by four.

LUA. See *LU Application*

LUA APIs. APIs provided to interface with and use the functions of logical unit (LU) programming. You can configure LUA APIs for all workstation connection types except asynchronous and coaxial (DFT).

LU Application. LUA shares a physical unit with the 3270 emulator, the gateway function, and the APPC API. LUA permits host attachment by the following: (1) Synchronous data link control (SDLC) (2) Token-ring network (3) X.25 (4) IBM PC network protocol (using a gateway) (5) Ethernet (ETHERAND) network protocol (using a gateway) (6) 3174 Peer Communications (7) Token-ring bus master (8) Integrated services digital network (ISDN) - SDLC - X.25 - IDLC

LU Name. The (fully-qualified) LU name contains both the network ID on which the partner LU resides and the partner LU name.

LU-to-LU Security. Protects your workstation from unauthorized network access. The security requires that only defined LUs that identify themselves by a password can establish communication sessions with your workstation. Once your workstation accepts the password, the local LU and the remote LU can begin a communications session. LU-to-LU security differs from conversation security that protects transaction programs (TPs)s. Only the LUs at each end of a session participate in LU-to-LU security. Intermediate nodes are not involved.

LZ9. A data compression method in which a coding dictionary is dynamically built during a session to substitute a single character for strings that are repeated. The method is typically applied with output codes of 9 bits, 10 bits, or 12 bits. LZ9 is sometimes referred to as an adaptive dictionary. If you are configuring a 3270 terminal or printer session (LU0, LU1, LU2, or LU3), LZ9 is required for data compression. The other compression levels do not allow data compression for 3270 terminal or printer sessions. See also *Compression Tokens*.

M

MAC. Media Access Control. The part of the data link layer that applies a medium access method. The MAC sublayer supports topology-dependent functions and uses the services of the physical layer to provide services to the logical link control sublayer.

Merlin. Working name for IBM OS/2 Version 4.

main window. See *primary window*.

master project. A project containing the actions that are inherited by all projects you create with the VRPG Client.

manipulation button. See *mouse button 2*.

maximize button. A button on the rightmost part of a title bar that a user clicks on to enlarge the window to its largest possible size. Contrast with *minimize button*, *hide button*.

media panel part. A part used to give the user control over other parts. For example, a media panel part can be used to control the volume of an audio part.

menu. A list of choices that can be applied to an object. A menu can contain choices that are not available for selection in certain contexts. Those choices are dimmed.

menu bar part. The area near the top of a window, below the title bar, and above the rest of the window that contains choices that provide access to other menus. In VRPG Client, you can drag a menu bar part from the parts palette or parts catalog and drop it onto a design window.

menu item part. A part that is a graphical or textual item on a menu. A user selects a menu item to work with an object in some way.

message. (1) Information not requested by a user but displayed by a product in response to an unexpected event or when something undesirable could occur. (2) A communication sent from a person or program to another person or program.

message file. A file containing application messages. The file is created from the message source file during the build process. See also *build*.

message subfile part. A part that can display predefined messages or text supplied in program logic.

migrate. (1) To move to a changed operating environment, usually to a new release or version of a system. (2) To move data from one hierarchy of storage to another.

.MID. The file extension of a MIDI file.

MIDI file. Musical Instrument Digital Interface file.

minimize button. A button located next to the rightmost button in a title bar that reduces the window to its smallest possible size. Contrast with *maximize button* and *hide button*.

mnemonic. A single character within the text of a choice identified by an underscore beneath the character. See also *mnemonic selection*.

mnemonic selection. A selection technique whereby a user selects a choice by typing the mnemonic for that choice.

mouse. A device with one or more push buttons used to position a pointer on the display without using the keyboard. Used to select a choice or function to be performed or to perform operations on the display, such as dragging or drawing lines from one position to another.

mouse button. A mechanism on a mouse used to select choices, initiate actions, or manipulate objects with the pointer. See also *mouse button 1* and *mouse button 2*.

mouse button 1. By default, the left button on a mouse used for selection.

mouse button 2. By default, the right button on a mouse used for manipulation.

mouse pointer. Synonym for *cursor*.

multi-line edit (MLE) part. A part representing an entry field that allows the user to enter multiple lines of text.

N

NAU Address. Allows you to indicate whether the local LU is independent or dependent. An Independent LU is not controlled by an SNA host, such as a S/390-based VTAM computer. A dependent LU NAU is controlled by an SNA host. If you select Dependent LU NAU, you need to specify (1) A network addressable unit (NAU) address and (2) A host link.

&periodNDF file. Node Definition File. The User's Guide is \CMLIB\CMVERIFY.txt.

NETID. See *Network ID*.

NetWare User ID. See *Local Node Name*.

Network ID. The name of your APPN network. The name of the network ID in which your local LU

resides. Also known as the SNA Network ID within Communications Manager. Within other IBM products, this name may be referred to as the NETID. Nodes within a network use the network ID as part of their node name. Together, the network ID and the node name (fully-qualified control point (CP) name) uniquely identify a node within an interconnected network environment. The network ID and the node name also uniquely identify a node for receiving error logs and network management alerts.

Network Management Agent. Novell's NetWare for Systems Application Architecture (NWSAA). The NMA allows a NetWare server to forward alerts to NetView and to receive and respond to certain requests from NetView.

Network Node. See *Network Node Server*.

Network Node Server. Node that can serve end nodes in its network with the full range of APPN services including Intermediate session routing, Directory services, and Topology and route selection services.

Network Node Server Address. 12-digit hexadecimal address for the LAN adapter on your network node server. May be in Ethernet or Token-Ring format and begin with a number ranging from 1 to 7.

NMA. See *Network Management Agent*.

node. An end point of a link or junction common to two or more links in a network. Nodes can be linked to host processors, communication controllers, cluster controllers, terminals, or workstations.

Node Type. (1) End node to network node server: your workstation uses a network node server. (2) End node - no network node server: your workstation does not have access to a network node server, or you do not know the address of your network node server. (3) Network node: your workstation serves as an APPN network node server. A network node Type 2 is usual for an OS/2 workstation LAN-attached to an AS/400 system.

notebook part. A graphical representation of a notebook. You can add notebook pages to the notebook part and then group the pages into sections separated by tabbed dividers. See also *notebook page part*, *notebook page with canvas part*.

notebook page part. A part used to add pages to a notebook part. See also *notebook*.

notebook page with canvas part. A combination of a notebook page part and a canvas page part. See also *notebook*, *canvas part*.

NTS/2. Network Transport Services/2.

O

object. (1) A named storage space that consists of a set of characteristics that describe itself and, in some situations, data. An object is anything that exists in and occupies space in storage and on which operations can be performed. Some examples of objects are programs, files, libraries, and folders. (2) A visual component of a user interface that a user can work with to perform a task. An object can appear as text or an icon.

object-action paradigm. A pattern for interaction in which a user selects an object and then selects an action to apply to that object.

object-oriented programming. A method for structuring programs as hierarchically organized classes describing the data and operations of objects that may interact with other objects.

object program. A target program suitable for execution. An object program may or may not require linking.

operating system. A collection of system programs that control the overall operation of a computer system.

outline box part. A part that is a rectangular box positioned around a group of parts to indicate that all the parts are related.

P

package. A function used to collect all the parts of a VRPG Client application together for distribution.

Partner LU Alias. An alternative name by which transaction programs may refer to the LU. In CM/2, the 5250 default is **5250PLU** (for instance, in PCS Setup).

Partner LU Dependent. The partner LU that you are communicating with does not support parallel sessions. Parallel sessions are two or more concurrently active sessions between the same two LUs. If Partner LU is dependent, the configuration is automatically updated to disable parallel session support.

Note: If you select Partner LU is dependent, you must enter a name in the Uninterpreted name field. If you do not select Partner LU is dependent, you cannot enter a name in the Uninterpreted name field. In this case, the uninterpreted name for that partner LU is automatically the same as the LU name.

Partner LU name. The fully-qualified name of the logical unit (LU) of the program or the host computer with which your emulator sessions communicate. A fully-qualified partner LU name consists of two parts

separated by a period (for example, TL377\$67.SY337777). The first part is the network ID of the host computer and the second part is the name of the LU. For an AS/400 system, the first part of the partner LU name is the same as **Local Network ID** and the second part is the same as the **Local Control Point Name** on the AS/400 system.

Partner Network ID. Name of the SNA workstation in which the partner node (the computer to which you are creating this peer connection) resides. Partner network ID + partner node name = control point (CP) of the computer that you want this connection to link you to. Connections that are established through your **network node server** do not require a partner network ID. Also known as the SNA Network ID within CM/2; within other IBM products, this name may be referred to as the NETID. If left blank in CM/2, defaults to the network ID that your local node resides in.

parts. Objects that make up the GUI of a VRPG Client application.

parts catalog. A storage space for all of the parts used to create graphical user interfaces for VRPG Client applications.

parts palette. A collection of parts that are most appropriate for building the current graphical user interface for an application. When you finish one GUI, you can wipe the palette clean and add parts from the parts catalog that you require for the next application.

pop-up menu. A menu that, when requested, appears next to the object with which it is associated. It contains choices appropriate for the object in its current context. You can drag a pop-up menu part from the parts palette or parts catalog and drop it onto a design window.

pop-up window. A movable window, fixed in size, in which a user provides information required by an application so that it can continue to process a user request. Synonymous with *secondary window*.

primary window. The window in which the main interaction between the user and the application takes place. Synonymous with *main window*.

programmable workstation (PWS). A workstation that has some degree of processing capability and that allows a user to change its functions.

progress indicator. One or more controls used to inform a user about the progress of a process.

project. The complete set of data and actions needed to build a single target, such as dynamic link library (DLL) or an executable file (EXE).

prompt. (1) A visual or audible message sent by a program to request the user's response. (2) A

displayed symbol or message that requests input from the user or gives operational information. The user must respond to the prompt in order to proceed.

pull-down menu. A menu that extends from a selected choice on a menu bar or from a system-menu symbol. The choices in a pull-down menu are related to one another in some manner.

push button part. A button labeled with text that represents an action that starts when a user selects the push button. You can drag a push button part from the parts palette or parts catalog and drop it onto a design window. See also *graphic push button part*.

PWS. Programmable workstation.

R

radio button part. A circle with text beside it. Radio buttons are combined to show a user a fixed set of choices from which only one can be selected. The circle is partially filled when a choice is selected. You can drag a radio button part from the parts palette or parts catalog and drop it onto a design window.

reference field. An AS/400 database field from which an entry field part can inherit its characteristics.

Remote SAP. Defines the address of the service access point (SAP) at the remote workstation that is used when communicating with your workstation. A hexadecimal value ranging from X'04' to X'9C' and divisible by four.

restore button. A button that appears in the rightmost corner of the title bar after a window has been maximized. When the restore button is selected, the window returns to the size and position it was in before it was maximized. *maximize button*.

S

SBCS. Single-byte character set

scroll bar. A part that shows a user that more information is available in a particular direction and can be moved into view by using a mouse or the page keys.

secondary window. A window that contains information that is dependent on information in a primary window and is used to supplement the interaction in the primary window. See also *primary window*. Synonym for *pop-up window*.

selection border. The visual border that appears around a VRPG Client part or a custom-made part allowing the part to be moved with the mouse or keyboard.

selection button. See *mouse button 1*.

server. A system in a network that handles the requests of another system, called a client.

server alias. A name you define that can be used instead of the server name.

settings notebook. A graphical representation that resembles a bound notebook containing pages separated into sections by tabbed divider pages. Turn the pages of a notebook to move from one section to another.

shared component. A component that can be accessed by more than one project.

single-byte character set (SBCS). A character set in which each character is represented by a one-byte code. Contrast with *double-byte character set (DBCS)*.

sizing border. The border or frame around a part (or set of parts) that you select to re-size the part (or set of parts) using the mouse or the keyboard.

slider part. A visual component of a user interface that represents a quantity and its relationship to the range of possible values for that quantity. A user can also change the value of the quantity. You can drag a slider part from the parts palette or parts catalog and drop it onto a design window.

slider arm. The visual indicator in the slider that a user can move to change the numerical value.

source directory. The directory in which all source files for a VRPG Client application are stored.

source part. A part that can notify target parts whenever the state of the source part changes. A source part can have multiple targets.

spin button part. A type of entry field that shows a ring of related but mutually exclusive choices through which a user can scroll and select one choice. A user can also type a valid choice in the entry field. You can drag a spin button part from the parts palette or parts catalog and drop it onto a design window.

SSCP. System Services Control Point.

SSCP-driven systems. VTAM host, S/3x0 computer, ES/9000-based computer, Other Communications Manager systems configured as gateways.

static text part. A part used as a label for other parts, such as a prompt for an entry field part.

status area. A part of a window that displays information indicating the state of the current view or object. See also *information area*.

Glossary

subfile part. A part used to display a list of records, each consisting of a number of fields. This part is similar to an AS/400 subfile. See also *subfile field*.

subfile field. A field used to define fields in a subfile part. See also *subfile part*.

sub-menu part. A part used to start a cascaded menu from a menu item or existing menu, or to start a pull-down menu from a menu item on a menu bar. See also *menu item part*.

T

tab stop. An attribute used to set a tab stop for a part so that users can focus on it when they use the Tab key to move through the interface.

target part. A part that receives a link event from a source part whenever the state of the source part changes.

target directory. The directory in which the compiled VRPG Client application is stored after a build. Contrast with *target folder*.

target folder. The object in which the icon representing a VRPG Client application is placed.

target program. The object to be built by the project, such as a dynamic link library (DLL).

thread. The smallest unit of operation to be performed within a process.

timer part. A part used to track the interval of time between two events and trigger the second event when the interval has passed.

title bar. The area at the top of each window that contains the system-menu symbol.

tool bar. A menu that contains one or more graphical choices representing actions a user can perform using a mouse.

topic. In dynamic data exchange (DDE), the set of data that is the subject of a DDE conversation.

tree view. A way of displaying the contents of an object in a hierarchical fashion.

U

Uninterpreted Name. Name that the partner uses in VTAM, where your partner LU is dependent and uses VTAM.

Unused Links. (In CM/2 DLC Adapter Parameters) APPC disconnects an unused link only when the maximum number of total links or the maximum number of incoming links is reached. Select **Free unused links** to have APPC disconnect a link as soon as the sessions using that link have ended.

user-defined part. A part consisting of one or more parts you have customized that you save to the parts palette or parts catalog for reuse. When in the palette or catalog, you can drag-and-drop this part onto the design window as you do any other VRPG Client part.

W

.WAV. The file extension of a wave file.

wave file. A file used for audio sounds on a waveform device.

window part. An area with visible boundaries that represents a view of an object or with which a user conducts a dialog with a computer system. You can drag a window part from the parts palette or parts catalog and drop it onto the project window.

window with canvas part. A combination of the window part and the canvas part. See also *window part* and *canvas part*.

work area. An area used to organize objects according to a user's tasks. When a user closes a work area, all windows opened from objects contained in the work area are removed from the workplace.

workplace. An area that fills the entire display and holds all of the objects that make up the user interface.

workstation. A device that allows a user to do work. See also *programmable workstation*.

X

X.25 Directory Entries. A list of X.25 directory entries you can use to create your connection. An X.25 directory is a table that associates user-provided names with the more detailed information required to make an X.25 connection between workstations. Use this field to select the X.25 directory entry you want to use for this connection.

Index

Numerics

5250LU 4, 70
5250PLU 4, 8, 9, 13, 14, 16, 17, 20, 70
OMXNZP 17

A

ADM 82
ADTS CS 3
Alias names
 Data area alias 15
 File alias 15
 Program alias 15
 Server alias 15
application
 install 26, 28
 maintain 29
 packaging 23
 VRAINST 28
AS/400 communications
 troubleshooting 76
AS/400 Information 58
 data area 86
AS/400 Information dialog 61
AS/400 Information notebook 15, 19
AS/400 local adapter address 4
AS/400 Local network ID 4
AS/400 logon information 21
AS/400 Network ID 10
AS/400 system 4

B

bibliography 229
blocking 69

C

Client Access/400 3, 9, 45
 Windows 3.1 36
CODE/400 3, 17, 80
Communications Manager 3, 6, 44, 47, 53, 56, 57, 58,
 59, 61, 62, 68
 5250LU 4, 70
 5250PLU 4, 8, 9, 13, 70
 FVDCEC 77
 FVDCERRS 77
 local LU 70
 partner LU 70
 Partner LU alias 4
 PLU alias 16
 PMDSPRAY 70
component
 installing 31

component (*continued*)
 packaging 25

D

database 14
DB2 68, 72
 Fixpak 37
 Windows 3.1 37
DBCS 83
Define Logon information 14
define server information 13
directories
 test 71
 WINTTEST 71
directory
 test 25
 Wintest 25

E

EVFCICFF 16, 43, 54
EVFICICFF 76

F

Fixpak 37, 59
FVDCEC 77
FVDCERRS 49, 77
FVDEPW.EXE 21
FVDERST.EXE 19

G

glossary 239

H

Heap 63

I

ICF device file 8, 14
Install.exe 33
installation
 ADTS CS 3
 component 31
 user-defined part 31
 variables 3
 VRAINST 28
 VRRINST 27

J

job description 14

L

LAN destination address 4
library list 14, 21
local adapter address 5
local control point name 4
local files 71
local LU 70
local network ID 4
local node name 4, 8, 10
LU alias 53
LU name 53, 56

M

message 86
mode name 4, 8

N

NetWare 45
network ID 8, 16

O

OS/2 3
OS/2 Configuration name 4
OS/400 communications 4
override 15, 19, 21

P

packaging 19, 23, 25, 26
partner LU 70
partner LU name 8
PC location name 4, 9
PC Support 9
PLU alias 8, 53, 58, 61, 62, 77
PMDSPY 70
PMDSPY 58, 61
Project 62, 74
 backup 73
 moving 75
 new 85
 rename 74, 85
 restore 74
 save 85
PTF 51

Q

QCMDDM 14

R

Remote location 4, 20, 21, 29, 54, 58
remote server table 15, 19, 20, 21
RST file 19, 20, 29, 58, 77
run time
 code 19

run time (*continued*)

 environment 19
 FVDEPW.EXE 21
 FVDERST.EXE 19
 install 26
 maintain 29
 override 19, 21
 packaging 23
 remove 30
 VRAINST 19
 VRRINST 19, 27
 VRRINST command 46
 Win-OS2 39
 Windows 3.1 33

S

Server alias 4, 15, 20, 58, 59, 61
Server Logon 62, 76
server name 4
service level 75
SQL 30, 72
 bind 72
Stack 63
STRCODE 13, 17, 43, 51, 54, 59, 60, 62, 68, 76
system to connect to 4

T

TPUPDATE 47, 57, 59
trace
 AS/400 system 77
 OS/2 78
 Windows 3.1 78

U

user profile 4, 10, 14, 21
user profile enrollment 10
user-defined part
 installing 31
 packaging 26

V

variables
 400011528646 4
 5250LU 4
 5250PLU 4, 8, 9, 16, 17, 20
 ALIAS01 16, 20
 AS400 4
 AS400DEV 4
 CAIBMOML 4
 OMXNZP 4
 QPCSUPP 4, 8
 TORAS016 4
 VRPGDEV1 4
VRAINST 19, 28, 56

VRPG Client/2 13, 17
VRPG project 62
VRRINST 19, 27, 46

W

WIN32S 27, 33, 39, 40, 51, 59, 68, 76
Windows 3.1
 Client Access/400 36
 DB2 37
 install 33
 memory considerations 35
 packaging 23
 restrictions 34
Windows 3.1 run time
 general differences 87
Workframe 47, 59, 63, 74



Printed in U.S.A.

S624-4841-00

